

Robots and Biological Systems: Towards a New Bionics?

NATO ASI Series

Advanced Science Institutes Series

A series presenting the results of activities sponsored by the NATO Science Committee, which aims at the dissemination of advanced scientific and technological knowledge, with a view to strengthening links between scientific communities.

The Series is published by an international board of publishers in conjunction with the NATO Scientific Affairs Division

A Life Sciences	Plenum Publishing Corporation
B Physics	London and New York
C Mathematical and Physical Sciences	Kluwer Academic Publishers Dordrecht, Boston and London
D Behavioural and Social Sciences	
E Applied Sciences	
F Computer and Systems Sciences	Springer-Verlag Berlin Heidelberg New York
G Ecological Sciences	London Paris Tokyo Hong Kong
H Cell Biology	Barcelona Budapest
I Global Environmental Change	

NATO-PCO DATABASE

The electronic index to the NATO ASI Series provides full bibliographical references (with keywords and/or abstracts) to more than 30000 contributions from international scientists published in all sections of the NATO ASI Series. Access to the NATO-PCO DATABASE compiled by the NATO Publication Coordination Office is possible in two ways:

- via online FILE 128 (NATO-PCO DATABASE) hosted by ESRIN, Via Galileo Galilei, I-00044 Frascati, Italy.
- via CD-ROM "NATO-PCO DATABASE" with user-friendly retrieval software in English, French and German (© WTV GmbH and DATAWARE Technologies Inc. 1989).



Robots and Biological Systems: Towards a New Bionics?

Edited by

Paolo Dario

ARTS Lab, Scuola Superiore S. Anna
56127 Pisa, Italy

Giulio Sandini

DIST, Università degli Studi di Genova
16145 Genova, Italy

Patrick Aebischer

Division de Recherche chirurgicale, Pav. 3
CHUV, 1011 Lausanne, Switzerland



Springer-Verlag Berlin Heidelberg GmbH

Proceedings of the NATO Advanced Workshop on Robots and Biological Systems,
held at Il Ciocco, Toscana, Italy, June 26–30, 1989

CR Subject Classification (1991): I.2.9, J.3

ISBN 978-3-642-63461-1 ISBN 978-3-642-58069-7 (eBook)
DOI 10.1007/978-3-642-58069-7

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1993
Originally published by Springer-Verlag Berlin Heidelberg New York in 1993
Softcover reprint of the hardcover 1st edition 1993
Typesetting: Camera ready by authors
45/3140 - 5 4 3 2 1 0 - Printed on acid-free paper

Preface

Defined as a kind of applied cybernetics, bionics evolved in the 1960s as a framework to pursue the development of artificial systems based on the study of biological systems. The primary goal of bionics, as defined by one of its pioneers, Henning E. Von Gierke, was “to extend man's physical and intellectual capabilities by prosthetic devices in the most general sense, and to replace man by automata and intelligent machines”. These objectives were pursued using models from the animal kingdom. For example, the ability of owls to swoop as quietly as they do, or the ability of beetles to create topographical maps of the terrain over which they fly, were examined. Numerous disciplines and technologies (some of which were still in their infancy), including artificial intelligence and learning devices, information processing, systems architecture and control, perception, sensory mechanisms, bioenergetics, etc., contributed to bionics research.

Attempts to develop intelligent machines by uncovering the principles underlying nature's examples proved to be more difficult than expected. In fact, the lack of implementation of some of the proposed models of intelligent biological systems into intelligent machines highlighted both the inadequacy of theoretical models alone, and the limitations of available technology.

This book contains revised papers originating from a NATO Advanced Research Workshop on Robots and Biological Systems, held in Il Ciocco, Italy, in June 1989, which was attended by about 60 scientists from 12 different countries. The purpose of the workshop was to explore the relationship between biological systems and robotic systems, and to discuss the question of whether the attempt to replicate the skilled behavior of biological systems in artificial systems has a better chance to be successful now than 30 years ago. The workshop was sponsored by the NATO Scientific Affairs Division, within the special program on Sensory Systems for Robotic Control.

The different perspectives proposed during the presentations stimulated many fruitful discussions and identified some strategic areas and future objectives for research on “new bionics”.

A consensus emerged on the value of the concept of "learning from nature" in order to derive guidelines for the design of intelligent machines (even if not incorporating anthropomorphic features) which operate in unstructured environments.

The significant progress in basic knowledge and technology that has occurred over the past 30 years has been recognized, and the need for continuous development in both areas has been indicated as a crucial factor for the fields of bionics and robotics. This research effort should be devoted on one hand to the understanding of the functions of biological systems, and on the other to the study of artificial sensory and motor subsystems and their coordination. Experimental robotics will play a crucial role in this respect.

However, the goal of designing and fabricating robotic systems with capabilities comparable to those of superior animal species remains elusive. Participants agreed that current science and technology would allow for the development of machines incorporating insect-like intelligence. Far from being reductive, this objective is significant since an insect-like autonomous robot would rely on advanced mechanical structures, sensors, actuators, control mechanisms, energy sources, etc., in order to be capable of intelligent sensory-motor behavior. Moreover, the exercise of assembling such insect-like robots as physical models of biological systems would allow for a verification of underlying design principles, as well as for an extensive testing of basic technologies and sensory-motor integration. Finally, insect-like micro-machines could have a number of useful applications in the field of advanced robotics, e.g., in monitoring and inspections.

In conclusion, the idea of revisiting the bionic approach to intelligent systems seems appropriate, especially in the light of present technological achievements and scientific knowledge.

The book comprises seven parts, each reflecting the objective of comparing the state-of-the-art in critical areas of biological and machine intelligence. The first three parts of the book are devoted to discussing sensory-motor aspects of vision, prosthetic hands and tactile perception, and legged locomotion. The fourth part presents a systematic comparison of intelligent motor control in biological and artificial systems, and also includes novel design concepts for actuation mechanisms. The fifth and sixth parts cover some technological aspects related to sensors, actuators, and interfaces between artificial devices and the nervous system. The final part explores the problem of cooperation among multiple units, and the emergence of collective intelligent behavior.

We wish to point out that the intensive work and fruitful discussions which were essential in achieving a successful workshop would have been impossible without the active and enthusiastic contribution of all participants. The environment at Il Ciocco also played a significant role in creating a relaxed and constructive atmosphere.

Ms. Lucia Lilli was very helpful in the editorial work associated with the assembly of this book.

January 1993

Paolo Dario
Giulio Sandini
Patrick Aebischer

Table of Contents

Part 1. Vision and Dynamic Systems

Active Perception and Exploratory Robotics	3
<i>R. Bajcsy</i>	
Object Identification and Search: Animate Vision Alternatives to Image Interpretation	21
<i>M.J. Swain, L.E. Wixson, D.H. Ballard</i>	
A Model of Human Feature Detection Based on Matched Filters	43
<i>M.C. Morrone, D.C. Burr</i>	
Visualizing and Understanding Patterns of Brain Architecture	65
<i>A. Rojer, E.L. Schwartz</i>	
Dynamic Vision	89
<i>H. Wechsler, L. Zimmerman</i>	
A Model of the Acquisition of Object Representations in Human 3D Visual Recognition	99
<i>S. Edelman, D. Weinshall, H.H. Bülthoff, T. Poggio</i>	

Part 2. Hands and Tactile Perception

The Perception of Mechanical Stimuli Through the Skin of the Hand and Its Physiological Bases	121
<i>R.T. Verrillo, S.J. Bolanowski, Jr.</i>	
Borrowing Some Ideas from Biological Manipulators to Design an Artificial One	139
<i>V. Hayward</i>	
Mechanical Design for Whole-Arm Manipulation	153
<i>W.T. Townsend, J.K. Salisbury</i>	
Whole-Hand Manipulation: Design of an Articulated Hand Exploiting All Its Parts to Increase Dexterity	165
<i>G. Vassura, A. Bicchi</i>	

Stable Grasping and Manipulation by a Multifinger Hand with the Capability of Compliance Control	179
<i>K. Tanie, M. Kaneko</i>	

Part 3. Locomotion

Mobile Robots – the Lessons from Nature	193
<i>D.J. Todd</i>	
Quadruped Walking Machine – Creation of the Model of Motion	207
<i>A. Morecki, T. Zielinska</i>	
Biped Locomotion by FNS: Control Issues and an ANN Implementation	223
<i>G.F. Inbar</i>	
How Fast Can a Legged Robot Run?	239
<i>J. Koechling, M.H. Raibert</i>	
Robot Biped Walking Stabilized with Trunk Motion	271
<i>A. Takanishi</i>	

Part 4. Intelligent Motor Control

A New Concept of the Role of Proprioceptive and Recurrent Inhibitory Feedback in Motor Control	295
<i>U. Windhorst</i>	
Analogic Models for Robot Programming	319
<i>R. Zaccaria, P. Morasso, G. Vercelli</i>	
Structural Constraints and Computational Problems in Motor Control	339
<i>F.A. Mussa-Ivaldi, E. Bizzi</i>	
Motion Control in Intelligent Machines	361
<i>A. Meystel</i>	
Control of Contact in Robots and Biological Systems	395
<i>N. Hogan</i>	
Motor Control Simulation of Time Optimal Fast Movement in Man	411
<i>B. Hannaford, L. Stark</i>	

Constraints on Underspecified Target Trajectories	419
<i>M.I. Jordan</i>	

Proposal for a Pattern Matching Task Controller for Sensor-Based Coordination of Robot Motions	445
<i>M. Brooks</i>	

Sensory-Motor Mapping with a Sequential Network	457
<i>L. Massone, E. Bizzi</i>	

Part 5. Design Technologies

Flexible Robot Manipulators and Grippers: Relatives of Elephant Trunks and Squid Tentacles	475
<i>J.F. Wilson, D. Li, Z. Chen, R.T. George, Jr.</i>	

Progress in the Design and Control of Pseudomuscular Linear Actuators	495
<i>G. Casalino, P. Chiarelli, D. De Rossi, G. Genuini, P. Morasso, M. Solari</i>	

Shape Memory Alloy Linear Actuators for Tendon-Based Biomorphic Actuating Systems	507
<i>M. Bergamasco, F. Salsedo, P. Dario</i>	

CCD Retina and Neural Net Processor	535
<i>A.M. Chiang</i>	

Retina-Like CCD Sensor for Active Vision	553
<i>G. Sandini, P. Dario, D.M. De Micheli, M. Tistarelli</i>	

Designing Artificial Structures from Biological Models	571
<i>M. Aizawa, H. Shinohara</i>	

Design Strategies for Gas and Odour Sensors Which Mimic the Olfactory System	579
<i>K.C. Persaud, J. Bartlett, P. Pelosi</i>	

Part 6. Interfacing Robots to Nervous System

Multi-Electrode Stimulation of Myelinated Nerve Fibers	605
<i>P.H. Veltink, J.A. van Alsté, J. Holsheimer</i>	

The Role of Materials in Designing Nerve Guidance Channels and Chronic Neural Interfaces	625
<i>R.F. Valentini, P. Aebischer</i>	
Regeneration-Type Peripheral Nerve Interfaces for Direct Man/Machine Communication	637
<i>G.T.A. Kovacs, J.M. Rosen</i>	
Integrated Bioelectronic Transducers	667
<i>M. Grattarola, A. Cambiaso, S. Cenderelli, S. Martinoia, M.T. Parodi, M. Tedesco</i>	
 Part 7. Robot Societies and Self-Organization	
A Robot Being	679
<i>R.A. Brooks, A.M. Flynn</i>	
Swarm Intelligence in Cellular Robotic Systems	703
<i>G. Beni, J. Wang</i>	
A Control Architecture for Cooperative Intelligent Robots	713
<i>J.S. Albus</i>	
Cellular Robotics – Construction of Complicated Systems from Simple Functions	745
<i>T. Fukuda, Y. Kawauchi</i>	
 List of Participants	 783

Part 1

Vision and Dynamic Systems

Active Perception and Exploratory Robotics

Professor Ruzena Bajcsy
Computer and Information Science Department
University of Pennsylvania
Philadelphia, Pennsylvania 19104

1 Introduction

Most past and present work in machine perception has involved extensive static analysis of passively sampled data. However, it should be axiomatic that perception is not passive, but active. Furthermore, most past and current robotics research use rather rigid assumptions, models about the world, objects and their relationships. It is not so difficult to see that these assumptions, most of the time, in realistic situations do not hold, and hence, the robots do not perform to the designer's expectations.

Perceptual activity is exploratory, which implies probing and searching. We do not just see, we look. We do not only touch, we feel. And in the course, our pupils adjust to the level of illumination, our eyes bring the world into sharp focus, our eyes converge or diverge, we move our heads or change our position to get a better view of something, and sometimes we even put on spectacles.

Similarly, our hands adjust to the size of the object, to the surface coarseness and to the hardness or compliance of the material. This adaptiveness is crucial for survival in an uncertain, and generally, unfriendly world as millenia of experiments with different perceptual organizations have clearly demonstrated. Although no adequate account or theory of activity of perception has been presented by machine perception research, very recently some researchers have recognized the value of actively probing the environment and emphasized the importance of data acquisition during perception including head/eye movement [3][7].

Because of the realization of today's inadequacies of robotic performances, we in the GRASP laboratory at the University of Pennsylvania for the past five years have embarked on research in Active Perception and Exploratory Robotics. What follows is an exposé of our theoretical foundation and some preliminary results. First, we shall describe what we mean by Active Perception, then we shall argue that Perception must also include manipulation, and finally, we will present Exploratory Robotics as a paradigm for extracting physical properties from an unknown environment.

2 What is Active Perception?

In the robotics and computer vision literature, the term “active sensor” generally refers to a sensor that transmits (generally electromagnetic radiation, e.g., radar, sonar, ultrasound, microwaves and collimated light) into the environment and receives and measures the reflected signals. We include under the term Active Perception, Active Sensing as well. We believe that the use of active sensors is not a necessary condition on active sensing, and that active sensing can be performed with passive sensors (that only receive, and do not emit, information), employed actively. Here we use the term active not to denote a time-of-flight type sensor, but to denote a passive sensor employed in an active fashion, purposefully changing the sensor’s state parameters according to sensing strategies. Putting it more succinctly, we are introducing a new paradigm for research in Machine Perception [4,5] called Active Perception. The new ingredients of this paradigm are taking multiple measurements and their integration, and the inclusion of feedback. Hence the problem of Active Sensing can be stated as a problem to control strategies applied to a data acquisition process that will depend on the current state of the data interpretation including recognition. The question may be asked: “Is Active Sensing only an application of Control Theory?” Our answer is: “No, at least in its simple version.” Here is why:

The feedback is performed not only on sensory data but on complex processed sensory data, i.e., various extracted features, including relational features. We do not have complete descriptions of the states of the system. Furthermore the models that are used here are a mixture of numeric/parametric and symbolic information.

But one can say that Active Perception is an application of intelligent control theory which includes estimation, reasoning, decision making and control [6]. This approach has been eloquently defended for Computer Vision by Tenenbaum [21]: “Because of the inherent limitation of a single image, the acquisition of information should be treated as an integral part of the perceptual process...Accommodation attacks the fundamental limitation of image inadequacy rather than the secondary problems caused by it.” Although he uses the term “Accommodation” rather than “active sensing”, the message is the same. Before we can outline the problem of active sensing more formally, we need to spell out the assumptions under which we are making the design.

The assumptions are that we have a priori available or we can extract:

1. Models of sensors and all subsequent processing modules, i.e., physics and geometry of the modules, including noise and uncertainty considerations.
2. The models of integration process of different modules, that is, combination rules and feedback.

3. Explicit specification of the initial and final state/goal and of the task.

If Active Perception is a theory, what is its predictive power? There are three components to our theory each with certain predictions:

1. Models at each processing level are characterized by parameters. These parameters are estimated using estimation theory and determine the lower bounds of performance.
2. The combination rules again predict the lower bounds of the final outcome from the system.
3. The task model and the final state/goal specification guarantees the termination of the process and predicts the cost of accomplishing the task.

2.1 The Models

When we speak about models of sensors we are not restricted to the hardware only but also include various software modules that play a role in the processing chain. The following highlights of this work are worth mentioning.

Sensory models:

1. Physics models. These models represent the mathematical equations of principles that the sensors operate. The analysis of these models provides range for expected performance of the sensors if no other influences than physics are at work. Examples of these models are optics, illumination, radiance, and forces.
2. Geometric models. Here we get predictions from various aspects of geometry on the best possible values. An example is the geometry of a pair of stereo cameras predicts how resolution decreases as a function of distance [19].
3. Ideal Measurement or Signal models. These models will help us analyze and predict the feasibility of detection of certain features. Examples of this case are: edge (step, linear or non-linear) and region (piece-wise constant, or linear or nonlinear, but monotonic) models [9, 17].
4. Noise or Disturbance models. Here we have considered not only the abnormal distribution (as everyone else has) but also abnormal distributions, symmetric or non-symmetric distributions of the random variables.

All these models provide upper and lower bounds for expected errors, resolution, and robustness, which is necessary for making certain decisions, in particular: "Do we need more data in order to get more accuracy? Can we

afford to take more data based on some economy? Given the errors how do we combine different pieces of information in order to improve the overall performance?" (For details, see Hager [11])

The Models and Estimation theory have been very successfully applied by Zucker in 1985 [23]. In this basic work titled "Theory of Early Orientation Selection", Zucker used the model of a contour that comes from differential geometry. He divides the orientation selection process into three steps: 1) The measurement step-series of convolutions. 2) The interpretation step of these convolution values. (This is a functional minimization problem.) 3) Finding the integral curve through the vector field. This decomposition into steps, having the parameters of each step explicit, allows Zucker to make clear predictions about where the contours will or will not be found. We very much agree with Zucker's criticism of the field for the lack of this kind of methodology! The very same flavor is in the paper of Leclerc and Zucker [14] where they study the edge detection of image discontinuities. The work of Binford and Nalwa [16] is again similar in flavor but applied to the modeling of edges or more general discontinuities.

2.2 A Concrete Example

A systematic and thorough approach to modeling, as it applies to Active Vision, is shown in the recent Ph.D. thesis of E. Krotkov [13] at the University of Pennsylvania. He has defined the task of determination of spatial layout using an agile camera system and two cues: range from focus and range from vergence. He has decomposed the problem into three subproblems: 1) Identifying an appropriate model M to represent the spatial layout of the environment; 2) finding effective methods for constructing M from vision data; and 3) determining strategies for actively, dynamically, and adaptively setting sensor parameters for acquiring the vision data.

In this section, we shall review only the first subproblem. Krotkov modeled two characteristics of objects – extent and position – in the environment. This means encoding a map of location of objects with respect to the viewer. In order to accomplish the above, he had to model the details of the sensor (the camera) as well as the details of the computational process of obtaining range from focus and range from vergence.

It is not possible to go into all the details of the analysis but we can summarize the model as follows:

1. determine the optics of the lenses, the depth of the field, the accuracy of object distance, (in this setup the distance of the object is independent of the depth of field for distances 1-3 m.)
2. circle of confusion; its diameter depends upon the distance of the object plane from focusing distance. For a given distance between the image and detector planes the confusion circle is directly proportional to the diameter of the aperture, in this case diameter is 58mm.

3. the spatial resolution of the detector array is another limiting factor; (for the CCD chip used in this work the width of one photoreceptor is 0.03 mm and the focal length $f=105\text{mm}$ determines the evaluation window size, typically 20×20 pixels).
4. determine how to measure the sharpness of focus with a criterion function. After analyzing defocus as an attenuation of high spatial frequencies and experimentally comparing a number of possible criterion functions the method based on maximizing the magnitude of the intensity gradient was chosen. It proves superior to others in monotonicity about the mode and in robustness in the presence of noise. Then the Fibonacci search technique is employed to optimally locate the mode of the criterion function.
5. the distance to an object point, given the focus motor position of sharpest focus is modeled by the thick lens law.

All the above predictions were experimentally verified on more than 3,000 points.

A very similar exercise that can be presented, although will not be for lack of space, is the modeling of the physical relationships for the vergence controller and the modeling of the line finder that is being used for matching the two stereo pairs of lines.

3 Perception Using Manipulation

The motivation for this approach is the observation that it is impossible to discern movable and removable object/parts without manipulating them. This problem is rather broad though fundamental in Perception. In order to make some progress, we have limited ourselves to a subproblem which is how to decide that two objects are detachable [22]. We postulate that this cannot be decided only by vision, or in general, by any noncontact sensing. An exception to this is the case when the objects/parts are physically separated so that the noncontact sensor can measure this separation or one knows a great deal of a priori knowledge about the objects (their geometry, material, etc.). We assume no such knowledge is available. Instead we assume that the scene is reachable with a manipulator. Hence, the problem represents a class of problems of segmentation that occur on an assembly line, bin picking, organizing a desk top, etc.

What are the typical properties of this class of problems?

1. The objects are rigid. Their size and weight is such that they are manipulable with a suitable end effector. Their numbers on the scene are such that in a reasonable time each piece can be examined and manipulated, i.e., the complexity of the scene is bounded.

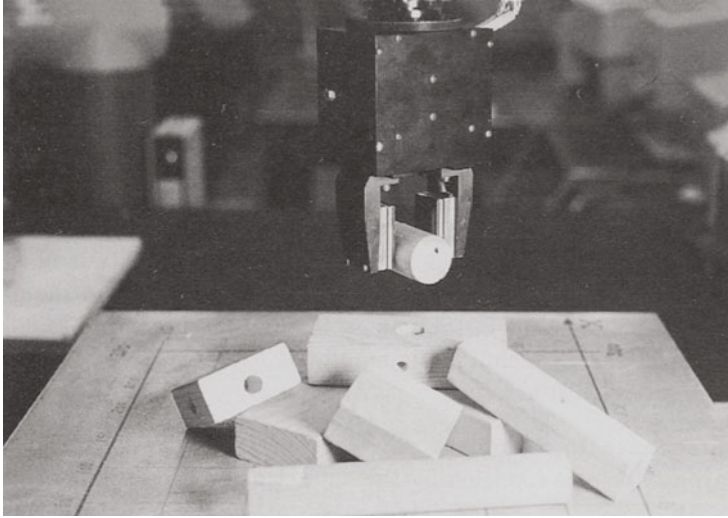


Figure 1: Objects to be segmented.

2. The scene is accessible to the sensors, i.e., the whole scene is visible, although some parts may be occluded, and reachable by the manipulator.
3. There is a well defined goal which is detectable by the available sensors. Specifically, the goal may be an empty scene, or an organized/ordered scene.

The segmentation problem as is specified above is a subclass of a more general problem of a disassembly task that we wish to address in the future. As for any perceptual theory, the theory of segmentation using manipulation must have the following components: models of sensors, world/scene models, task/utility models, and models of actions. The segmentation process is formulated in terms of graph-theoretic operations that are mapped into corresponding manipulatory actions.

1) **Models of sensors:** these include the characterization of the non-contact sensor such as the spatial resolution, signal to noise ratio, the physical parameters of the different end effectors, such as the vacuum suction

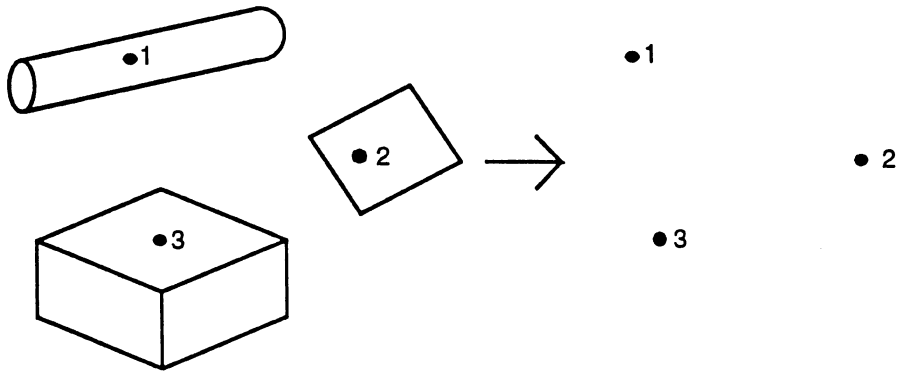


Figure 2: An example of a graph of a dispersed scene.

cup, the size of the spatula for pushing objects, the span of the gripper, and the maximum allowable weight and/or force.

2) **Models of objects:** specified in terms of their geometry, size and substance.

3) **The Model of our world:** this work is limited to arrangement of objects thrown at random on a plane, called a heap. Then a scene is a (partial) view of a heap. The objects in the scene are represented as nodes in the digraph and the arcs denote: on-top-of relation. It is important to emphasize that this digraph represents relations of only the visible surface segments, i.e., as they appear through the visual sensor which is not always the same as the physical objects and their surface segments. The true physical arrangements of objects on the scene as well as the part-whole relations of objects are not known.

The scene can be classified based on the analysis of the digraph into the following categories: Empty, if there are no nodes in the graph; Dispersed, if there no arcs in the graph; Ambiguous, if there is a cycle in the graph; Overlapped, if there are at least two nodes connected with one arc in the graph; and Unstable – this category is not tested by the analysis of the graph but through analysis of the contact point/line of the object with the support plane. If this contact is point or line it is classified as unstable. Figure 2 displays an example of a graph of a dispersed scene.

4) **Task models:** The final goal of the process. An example of a final goal can be the empty scene and the intermediate goals then can be those scenes

that are more simply measured by a cost/benefit function. This cost/benefit function entails the cost of performing the particular manipulation, and the benefit is measured via the estimate of the outcome of the manipulation with respect to the final goal, i.e., emptying the scene.

5) Models of Action: Parametrises the scene/object/manipulation interaction.

In principle there are two kinds of Actions: 1. Sensing action, i.e., data acquisition action (look and/or feel), and 2. Manipulatory action.

The purpose of the manipulatory actions for this paper is to exhaust a physical disturbance, being either global (as shaking is) or local (a pushing/pulling). In view of our formulation of the segmentation problem as a graph generation/decomposition problem we classify the manipulatory action in relationship to the operations that apply on the digraph. There are two such operations: the node removal, which means in terms of manipulation the removal of an object from the scene, and the arc removal which in turn translates into object displacement in the scene so that the relationship of on-top-of does not hold anymore between the two objects. Putting it another way: an isomorphism exists between the manipulation actions and graph decomposition operations. Our approach is to close the loop between sensing and manipulation. The manipulator is used to simplify the scene by decomposing the scene into visually simpler scenes. The manipulator carries the contact sensors to the region of interest and performs the necessary exploratory movements that will determine the nature of the mechanical binding between objects in the region. Perception-Action interaction is modeled by a non-deterministic finite state Turing Machine. The model of sensing, manipulation and control is a Non-deterministic Turing Machine (NDTM) as we show in Figure 3. The physical world (scene) is the “tape” of the machine, the “read_from_tape” actions are the sensing actions and the “write_to_tape” actions are the manipulation actions. The model is a Turing machine because the manipulation actions constantly change the physical environment (tape) and hence its own input. The above model is non-deterministic because of the non-predictable state of the scene after each manipulatory step. From this, of course, follows also the non-deterministic control of actions. In addition to the non-determinism of the control strategies, the automaton has finite states, which are determined by the finite numbers of recognizable scenes and the finite number of available actions.

We believe that this model is quite general providing that one can quantize the scene descriptions and/or the sensory outputs into unique and mutually exclusive states, and, of course, one has only a finite number of manipulatory actions.

There are several advantages to the formalisms of the non-deterministic finite state Turing machine. The first advantage [1] is that the sense-compute-act formalism allows the control problem to be partitioned in time and com-

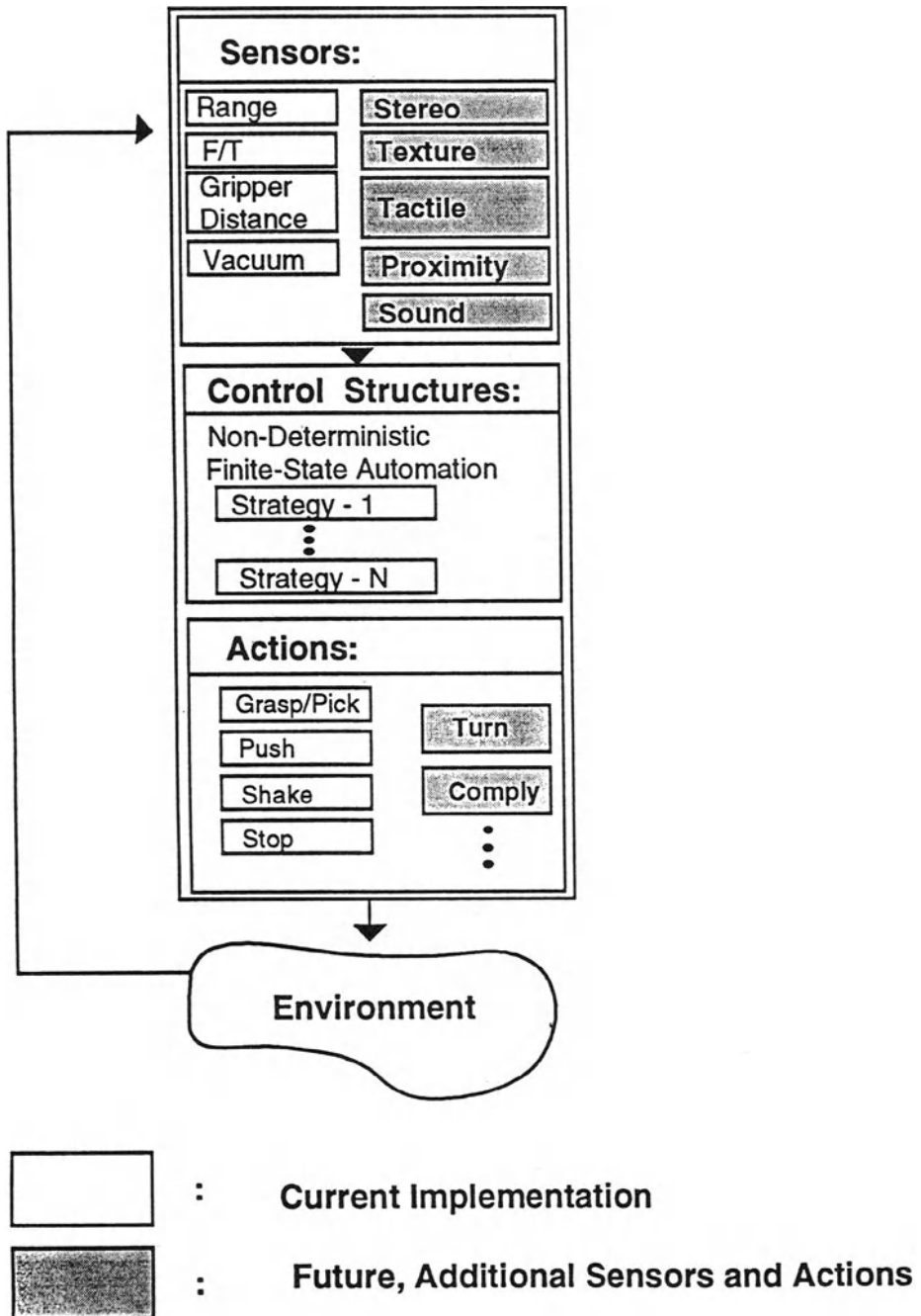


Figure 3: The Non-Deterministic Turing Machine (NDTM)

plexity. At any given time, the system deals only with present state and present input, produces an output which is a function of current state and current input and moves to a new state. The current state encodes information about the past history of states and actions of the machine and its environment. Current sensory input is not deterministic (noise in sensory data). The next state of the NDTM is not deterministic because the machine modifies its tape via actions whose outcome cannot be known a priori (push and shake actions).

The second advantage is that the theoretical tools needed to prove correctness of the machine's behavior have long been established and tested. Path sensitization and graph de-cyclization algorithms exist, [10, 12, 8] to prove: 1) The goal state is reachable and 2) The state transition diagram does not contain deadlock states, or cycles.

The third advantage is that it facilitates error handling. If additional states need to be defined to deal with non-anticipated error conditions, then these states can be simply inserted.

The fourth advantage is that it is modular and allows insertion of new sensors, actions and feedback conditions.

The fifth advantage is that it makes debugging easy.

The sixth advantage is that it allows a system to be developed incrementally.

One disadvantage is that the number of states and transitions needed to represent the machine and its environment increases as more sensors are added. Addition of more sensors implies increased complexity.

4 Exploratory Robotics

Much of the work in Robotics until now has been by and large conducted in the so-called "knowledge driven" framework. The justification for this approach was the fact that in the industrial environment the geometry, material, environmental conditions and the task are (1) quite constrained, (2) known a priori, and (3) well controllable. However, this is not the case in many other situations and applications of robots in underwater, mine and outer space explorations. The common denominator to all of these cases is that the robot must be able to explore and adapt to unconstrained and unknown environments. This is the motivation for the investigation of Exploratory Robotics.

4.1 Definition of the Problem

We wish to investigate the necessary components/modules that must be embedded into a robot with Exploratory Capabilities. These ideas came from our collaborative efforts between R. Klatzky and S. Lederman, see [15]. In

other words, what sensors, exploratory procedures, data processing, data reduction and interpretation capabilities for a given task must such a robot have. In full generality, this task is formidable. Hence, we shall limit ourselves to two more specific tasks: 1) Exploration of surface properties of ground for mobility purposes, and 2) Exploration of an object for manipulatory purposes.

In the first task, we shall consider surfaces made from materials such as dirt/soil/sand, rocks/concrete, pebbles/gravel, metals, wood, glass/ceramics, rubber/polymers, and viscous mixtures (like mud). We shall not consider vegetables, textiles, liquids, and like materials.

In the second task we will limit ourselves to objects by size and weight. This limitation will be determined by the size and flexibility of the end-effector, i.e., we shall consider objects that are graspable. This will exclude liquid, for example, but not deformable objects like a cable or a rubber ball. We shall also investigate objects that have two rigid parts joined by a hinge.

For both of the tasks, the robot will be equipped with one six-degree freedom manipulator and a range finder and/or a pair of CCD cameras, called the *LOOKER*, and ONE six-degree freedom manipulator and a hand, called the *FEELER*. The *LOOKER*, depending on the need, can also have a color camera system or any non-contact electromagnetic wave measuring detector (infrared is one possibility). The *FEELER* has a force/torque sensor in its wrist and a hand with three fingers and a rigid palm. Each finger has one and one-half degrees of freedom.

The sensors on the hand include: a position encoder and force sensor at each joint of the finger; a tactile array at each of the finger tips and on the palm; a thermo-sensor on the palm, and ultrasound sensor on the outer side of the hand. In addition the hand has available various tools that it can pick up under its control.

Both the *FEELER* and *LOOKER* are under software control of strategies for data acquisition and manipulation. For the first task, we consider a model of a foot with a planar sole as one tool that will act as the probe for testing the surfaces for mobility.

4.2 Exploration of Surface Properties

The Problem

Given a surface, we wish to establish procedures to determine physical and geometric properties with minimal a priori information so that an object like a robot or vehicle can move on this surface. The basic assumption is that the surface is much larger than the robot and, at least, locally flat so that there is space to move around. The flatness assumption is relative to the size of the robot: the surface variation from a planar surface must be no more than 10% (3dB) of the sole of the robot's foot. We do not consider the problem of obstacles.

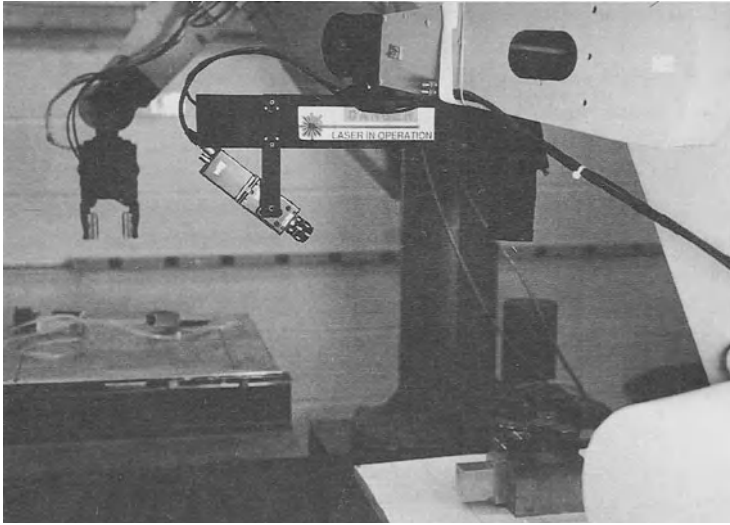


Figure 4: The *FEELER* and *LOOKER*.

Scientific fields older than robotics have investigated how to measure the attributes of the materials listed in Table 1. They are mineralogy, geology, soil science, civil engineering (for testing soil in preparation for building), and material sciences in general. Tests from these fields share the following procedures: (1) take samples into the laboratory and perform a multitude of tests, (2) if necessary, perform destructive tests, such as for brittleness or penetrability, or even for deformability, and (3) perform excavations of layered surface (as in geology).

The question for this research is which of these procedures are applicable for our domain. The procedure in (1) can be applied in the robotic context. One can design a robot in such a way that it can carry with it a small testing kit. The point in (2) is harder to envision though as part of the calibration process can be executed. The last point, (3), is totally inapplicable since the robot will not have time to perform excavation before it moves.

We examine those Exploratory Procedures (EPs) which will allow the robot to: a) stand firmly on the surface (static stability) and b) move on the surface in a stable manner (dynamic stability).

MATERIALS AND THEIR SALIENT ATTRIBUTES								
ATTRIBUTES	CLASSES OF MATERIALS							
	Metals	Rocks Concrete	Glass Ceramics	Rubber Polymers	Wood	Soil Sand	Pebbles Gravel	Viscous Mixtures
Penetrability	No	No	No	No	No	Yes	Yes	Yes
Deformability	No	No	No	Yes	No	Yes	Yes	Yes
Hardness	Yes	Yes	Yes	Yes	Yes	No	No	No
Brittleness	No	Yes	Yes	No	No	No	No	No
Compressibility	No	No	No	No	No	Yes	Yes	No
Compressive Strength	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Surface Roughness	Yes	Yes	No	Yes	Yes	Yes	Yes	No
Thermal Conductivity	Yes	No	Yes	No	No	No	No	No
Electrical Conductivity	Yes	No	Yes	No	No	No	No	No
Magnetic Permeability	Yes	No	Yes	No	No	No	No	No
Optical Properties	Yes	No	Yes	No	No	No	No	No
Viscosity	No	No	No	No	No	Yes	Yes	Yes

Yes - Attribute exists, is measurable and is a distinguishing property.

No - Attribute does not exist or is not measurable or is not a distinguishing property.

Table 1: Materials and their salient attributes

Further Assumptions

In order to further constrain the interpretation of the measurements we eliminate the effect of the geometry, that is, we assume that both the *LOOKER* and the *FEELER* are perpendicular to the examined surface.

Exploration for static stability: Exploratory Procedure for surface firmness versus penetrability. The penetrable surface can be deformable, compressible, either, or both. As an example, whereas penetrable objects such as soil, sand, pebbles, viscous mud, and rubber/polymers are deformable; only soil, sand, and pebbles are compressible (see Table 1).

This EP will utilize a cooperative effort between vision and force guided penetration. The *FEELER* exerts controlled and recorded force on the surface while the *LOOKER* observes the surface. If the surface has not changed under the given force, then it is firm; if it deforms then it is penetrable. It can be either deformable or compressible. The test for discriminating the latter two is to use the *LOOKER* observation of the resulting surface after the *FEELER* has withdrawn the penetrating force. If the surface has not changed from the previous image then we have a deformable surface (just like mud would stay); otherwise we have a compressible surface. Naturally, this is not a sufficient test, especially when the measurement indicates no firm surface. Other tests like measure of pressure, surface roughness and viscosity must be carried out. Which ones are necessary and sufficient will be one of the topics of this research.

4.3 Exploration of Graspable Objects

The Problem

We wish to find the following properties of the graspable object: material (its hardness and surface texture), density, temperature, weight and size, rigidity versus flexibility, and finally gross shape for identifying graspable points.

In order to accomplish this task one needs two modes of exploration: a Static mode and a Dynamic mode. In the Static mode the object is stationary and the *LOOKER* and the *FEELER* can *look* and *feel* around the object. During the Dynamic mode the object is being grasped and manipulated, for example lifted or shaken. In the Static mode we can establish the following attributes: size, shape, temperature and hardness/surface texture. In the dynamic mode the remaining attributes are established: the weight, density and the rigidity versus flexibility.

The Static Exploratory Procedures applied on objects

Following the work of Allen [2] and Stansfield [20] we accept their findings that blind touch is unproductive and the tactile exploration should be guided

by vision. Hence we begin with the *LOOKER* which will give us the position, gross shape and size of the object. Using the superquadric fitting to the visual three-dimensional data developed by Solina [18], we get further parametrization of the data, that is: the orientation, extent in three orthogonal planes (the size), and estimate of the surfaces (whether they are planar or second order surfaces) of the object. Then following Stansfield's EPs for hardness and surface texture and using the *FEELER* we can estimate the material of the object. In addition, by measuring the conductivity of the material (by another similar low level EP), we can further distinguish the material as metal or non-metal. All these properties are passed to the next stage – the Dynamic mode.

The Dynamic Exploratory Procedure applied on objects

As mentioned before, the dynamic EPs will measure weight, density and rigidity. EPs for weight and density: Grasp the object and lift it to a height H . The exerted force divided by approximately .9 (gravitational force) will give the weight of the object. The weight divided by the volume (calculated from the shape parameters) is the density of the material.

The more sophisticated Exploratory Procedure is the test for rigidity. Another assumption: consider objects either rigid, bent, or as two parts connected with a hinge. This again involves a cooperation between the *LOOKER* (the vision) and the *FEELER* (with force-guided probe). There are several strategies that must be followed in a few specified orders:

1. Consider an object which is being translated or rotated on the table by pushing (we know the magnitude and direction of the exerted force). If the new image can be accounted for by rigid transformations for this manipulation, then the object is rigid; otherwise the change must be examined.
2. Examination of the change: parts are rigid but their spatial relationship has changed, or the object is bent, i.e., a deformation has occurred.
3. The case of rigid parts indicates that there is either one fixed point of rotation, or one fixed line of rotation. In either case we have identified a hinged object.
4. In the case of a bent object, compute the amount of bend.

5 Conclusion

We have defined Active Perception as a problem of an intelligent data acquisition process. For this, one needs to define and measure parameters and errors from the scene which in turn can be fed back to control the data acquisition process. This is a difficult though important problem. Why? The difficulty

is that many of the feedback parameters are context- and scene-dependent. The precise definition of these parameters depends on a thorough understanding of the data acquisition devices (camera parameters, illumination and reflectance parameters), algorithms (edge detectors, region growers, 3D recovery procedures) as well as the goal of the visual processing. The importance, however, of this understanding is that one does not spend time on processing and artificially improving imperfect data but rather on accepting imperfect, noisy data as a matter of fact and incorporating it into the overall processing strategy.

The second point we made is that manipulation is an essential part of the perceptual process. The hand is as the eye: a sensory device. Subsequently, one needs to consider not only signal processing modules but also basic manipulatory action called exploratory procedures as an essential ingredient of perceptual theory.

The third and last point we are making is a case for Exploratory Robotics. Today, it is assumed that the size and shape of the object is sufficient for grasping purposes. It should be very apparent that unless one knows what materials are being used the system may be easily fooled. And even if we know the material of the outer surface, we do not know the inside, which may very dramatically change the weight, and hence, the grasping strategy. Our research aims to fill this gap. The question of rigidity is also very crucial when a grasping strategy is considered. Furthermore, the tests for hinges and bending are the first tests towards testing the functionality of an object. In the test for rigidity, we need to further explore what changes will occur when other controlled manipulatory actions will be applied on such objects, for example, lifting or rotating the object in space. All these steps are part of a general examination of the object, finding stable positions, etc. All these tests lead to understanding of what the necessary components are for a general purpose Perceptual Theory.

Acknowledgements

This research was funded in part by the United States Postal Service, BOA Contract: 104230-87-H-0001/M-0195, DARPA/ONR grant N0014-85-K-0807, NSF grant DCR 8410771, Air Force grant AFOSR F49620-85-K-0018, Army grant DAAG-29-84-K-0061, by DEC Corporation, IBM Corporation, and LORD Corporation.

References

- [1] Albus, J., Barbera, A. & Fitzgerald, M. (1982) *Programming a Hierarchical Robot Control System*, Proceedings of the 12th International Conference on Industrial Robots, Paris, France.

- [2] Allen, P.K. (1987) *Robotic Object Recognition Using Vision and Touch*, Kluwer, Norwell, MA.
- [3] Aloimonos, J. & Badyopadhyay, A. (1987) *Active Vision*, Proceedings of the 1st IEEE International Conference on Computer Vision, London, U.K.: Computer Society Press, pp. 35-54.
- [4] Bajcsy, R. (1985) *Active Perception vs. Passive Perception*, Proceedings of the 3rd Workshop on Computer Vision: Representation and Control Bellaire, MI., Computer Society Press, pp. 55-59.
- [5] Bajcsy, R. (1988) *Active Perception*, Proceedings of the IEEE, 76, pp. 996-1005.
- [6] Bajcsy, R., Gupta, A. & Mintz, M. (1989) *Research on Symbolic Inference in Computation Vision*, Technical Report, University of Pennsylvania.
- [7] Ballard, D.H (1987) *Eye Movements and Spatial Cognition*, Computer Science Dept. Univ. of Rochester, TR 21, November 1987.
- [8] Deo, N. (1974) *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, Englewood Cliffs, NJ 07632.
- [9] Haralick, R.M. (1981) *The Digital Edge*, Proceedings of IEEE Computer Conference Pattern Recognition and Image Processing, Computer Society Press, pp. 285-291.
- [10] Hartmanis, J. & Stearns, R. (1966) *Algebraic Structure Theory of Sequential Machines*, Prentice-Hall, Englewood Cliffs, NJ 07632.
- [11] Hager, G.D. (1988) *Active Reduction of Uncertainty in Multi-Sensor Systems*, Ph.D. Dissertation, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA 19104.
- [12] Kohavi, Z. (1970) *Switching and Finite Automata Theory*, McGraw-Hill, New York.
- [13] Krotkov, E. (1987) *Exploratory Visual Sensing for Determining Spatial Layout with an Agile Camera System*, Ph.D. Dissertation, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA 19104.
- [14] Leclerc, Y.G. & Zucker, S.W. (1989) *The Local Structure of Image Discontinuities in One Dimension*, IEEE Transactions on PAMI, Vol. 9, No. 3, pp. 341-355
- [15] Lederman, S.J. & Klatzky, R.L. (1987) *Hand Movements: A Window into Haptic Object Recognition*, Cognitive Psychology, 19, pp. 342-368.

- [16] Nalwa, V.S. & Binford, T.O. (1986). On Detecting Edges. IEEE Transactions on PAMI, Vol. 8. pp. 699-714.
- [17] Pavlidis, R and Liou, Y.T. (1988) *Integrating Region Growing and Edge Detection*, submitted to *ICVPR*.
- [18] Solina, F.(1987), *Shape Recovery and Segmentation with Deformable Part Model*, Ph.D. Dissertation, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA 19104.
- [19] Solina, Franc (1985) *Errors in Stereo Due to Quantization*, GRASP lab. memo, Univ. of Pennsylvania, Philadelphia, December 1985.
- [20] Stansfield, S.A. (1988) *A Robotics Perceptual System Utilizing Passive Vision and Active Touch*, International Journal of Robotics Research, 7, pp. 138-161.
- [21] Tenenbaum, J. M. (1970) *Accommodation in Computer Vision* Stanford University Ph.D. Thesis.
- [22] Tsikos, C. (1987) *Segmentation of 3-D Scenes Using Multi-Modal Interaction Between Machine Vision and Programmable, Mechanical Scene Manipulation*, Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104.
- [23] Zucker, S.W. (1985) *Early Orientation Selection: Tangent Fields and the dimensionality of their support*, TR-85-13-R, Computer Vision & Robotics Laboratory, Dept. of EE, McGill Univ. Montreal, Quebec, Canada.

Object Identification and Search: Animate Vision Alternatives to Image Interpretation

Michael J. Swain, Lambert E. Wixson, and Dana H. Ballard

Computer Science Department
University of Rochester
Rochester, NY 14627, USA

swain@cs.rochester.edu

Abstract. We are accustomed to thinking of the task of vision as being the construction of a detailed representation of the physical world. However, a paradigm that we term animate vision argues that vision is more readily understood in the context of the tasks that the system is engaged in, and that these tasks may not require elaborate categorical representations of the 3-D world. As an example, we show how the general problem of image interpretation can be replaced in many cases by a combination of two simpler problems, identification and search. Both tasks use multidimensional color histograms to represent the model and images. Color histograms are shown to permit efficient matching and a sufficiently rich representation to distinguish among a large number of objects.

1 Introduction

We are accustomed to thinking of the task of vision as being the construction of a detailed representation of the physical world that can be used in the execution of various robotic tasks. Often this detailed representation is assumed to be one in which every object has been identified; this is the goal of a process commonly referred to as *image interpretation*. In contrast to this view, however, is a paradigm that we term animate vision. Animate vision argues that the representation needed for a given task is often considerably simpler than the elaborate categorical representation that is the goal of image interpretation. In many cases, the reason that a simpler representation suffices is related to possession of a mobile camera platform [Ballard, 1989]. This paper argues that while image interpretation has proven to be extremely difficult and computationally complex, there exist two easier tasks, search and

identification, that can often achieve goals that in a traditional view would be handled by an image interpretation system. Novel implementations of these tasks are presented which, to achieve fast execution times, rely heavily on color rather than shape to perform object recognition.

2 Vision as a Collection of Task-Specific Processes

One motivation for the animate vision view of vision as a collection of task-specific processes comes from the major functional divisions found in human and primate brains. A significant feature of the gross organization of the primate visual brain is the specialization of the temporal and parietal lobes of visual cortex [Mishkin and Appenzeller, 1987; Maunsell and Newsome, 1987]. The parietal cortex seems to be subserving the management of locations in space whereas the temporal cortex seems to be subserving the identification of objects in the case where location is not the issue. In a striking experiment by Mishkin [1987], monkeys with parietal lesions fail at a task that requires using a relational cue but have no trouble performing a very similar task that requires using a pattern cue. The reverse is true for temporal lesions. Why should the primate brain be specialized in this way? If we think generally about the problem of relating internal models to objects in the world, then one way to interpret this “What/Where” dichotomy is as a suggestion that image interpretation, the general problem of associating many models to many parts of the image simultaneously, is too hard. In order to build vision systems that are computationally tractable within a single fixation, i.e. that function in real-time, perhaps the problem must be simplified.

The goal of image interpretation is to identify and compute the pose of the objects in the image and from this data to construct a world model for future reference. When we consider applying a What/Where simplification, a decomposition into two tasks suggests itself. The first, *object identification*, attempts to identify a given portion of an image. The image portion is assumed to come from some sort of perceptual grouping, such as a segmentation arising from depth, motion, or intensity cues. The second task, *object search*, given a certain type of object to search for, attempts to search for an object either within an image or, more generally, within the world (by using a mobile camera like that described by Brown

		Object to Match Against	
		One	Many
Image Portions	One	Manipulation: trying to do something with an object whose identity and location are known	Identification: trying to identify an object whose location can be fixated
	Many	Search: trying to find a known object that may not be in view	Image interpretation: too hard?

Table 1: The biological organization of cortex into What/Where modules may have a basis in computational complexity. Trying to match a large number of image segments to a large number of models at once may be too difficult.

[1988]). Table 1 summarizes the visual tasks that arise from the possible combinations of the What/Where dichotomy.

A robot equipped with these two simpler capabilities does not need to resort to image interpretation for most applications. To see this, let us consider four common reasons why a robot might “decide” that the image produced by its camera(s) should be run through an image interpreter. One reason might be that something has attracted the robot’s interest to the image. Since it has not yet identified the objects in the image, this “attraction” must stem from some more primitive perceptual property of the scene, such as a particular texture, a certain arrangement of lines or surfaces, or the presence of motion. It is almost always the case, however, that in such cases the only object that it is really important to recognize is the object that gave rise to the interesting perceptual property. Since we know the portion of the image that gave rise to this property, an object recognition system only needs to be executed on this portion, not on the entire image as with image interpretation. This is exactly the role of the object identification process described above.

It might be argued that this rationale may not always work. For example, perhaps nearby motion attracted our interest and we identified the moving object as a bird, but what we are most concerned with discovering is whether the bird was startled by an approaching tiger. In this case, we can still avoid interpreting the entire image by using another member of our repertoire of visual processes, object search. This capability allows us to search within the image or to deploy our camera with the explicit purpose of finding a tiger.

A third reason that an image interpretation module might be used is that in many cases an ambiguous object can be identified better if neighboring objects are also being identified at the same time and if the labels attached to these neighbors can provide “context”. This approach is often formulated within paradigms which involve evidence weighting, constraint satisfaction, or energy minimization. An example of this might be attempting to identify an object which the system thinks is either a workstation or a terminal. In this case, the presence of a mouse near the object would provide strong evidence for the workstation interpretation. We hypothesize, however, that most of the benefits of this sort of contextual evidence can be easily produced by using the object search mechanism. A simple method might be to attach information about supporting objects to each object model. Thus, the workstation model might have an “annotation” which states that a mouse provides good evidence for this model. When the object identification process results in ambiguity, the search mechanism would set out to look for objects which can resolve the ambiguity. Thus, in our example, the search mechanism might search for a mouse in the area near the object.

Finally, probably the most prevalent reason for using an image interpreter to construct a detailed map is to remember the locations of objects for future reference. There are several problems with this view, however. The world may change, causing the representation to become invalid. A more serious problem, however, lies with the choice of the coordinates with which to denote the locations of the object in the world model; world coordinate systems are often impractical due to sensor and effector error [Brooks, 1987]. A fast object search process, however, obviates the need for such record keeping. Instead of remembering the exact location of the objects you have seen, simply search for objects as you need them. Ballard [1989], examining fixation traces of subjects instructed to remember the position of objects in a room, has conjectured that humans do not construct a detailed world model routinely. It is tempting to speculate that humans compensate by using their object search abilities.

The above four case studies demonstrate how two animate vision processes, object identification and object search, can achieve the same purpose as image interpretation. Note that we are certainly not claiming that all applications can avoid image interpretation. Robot applications such as mapping, whose goal is to construct a detailed world model, may require this capability. Our claim is simply that most robot applications do not require a

world model in which the individual objects are identified and their locations stored. In these situations, image interpretation is overkill.

In the remainder of the paper, we present two simple but novel methods for identification and search. Since both of these algorithms rely on color extensively, we turn first to a discussion of color, color spaces, and color histograms.

3 The Role of Color in Visual Processes

Color has been neglected recently as an identification cue, although it has been used in earlier work [Feldman and Yakimovsky, 1974; Garvey, 1976; Ohlander *et al.*, 1978]. One reason for this may have been the lack of good algorithms for color constancy. However, recently there has been great progress in correcting for both the chromaticity of the illuminant [Maloney and Wandell, 1986; Forsyth, 1988; Rubner and Schulten, 1989; Hurlbert and Poggio, 1987] and for geometric effects such as specularities [Klinker *et al.*, 1988; Bajcsy *et al.*, 1989; Healey and Binford, 1987]. Given that reasonable color constancy can be achieved, color has enormous value in vision because it is a local surface property that is view invariant and largely independent of resolution. Shape cues, by contrast, are highly resolution dependent, and only a highly restricted set are view invariant (e.g. corners, zeros of curvature).

Perhaps another reason that color has not been used is that it is not intrinsically related to the object's identity in the way that other cues, e.g., form, are. This view is well represented by Biederman [1985]:

“Surface characteristics such as color and texture will typically have only secondary roles in primal access ... we may know that a chair has a particular color and texture simultaneously with its volumetric description, but it is only the volumetric description that provides efficient access to the representation of CHAIR.”

However, this opinion is easily challenged. There are many examples from nature where color is used by animals and plants to send clear messages of enticement or warning. The manufacturing sector uses color extensively in packaging to market goods. Animate vision

systems can also use representations that are heavily personalized to achieve efficient behaviors. For example, it may not be helpful to model coffee cups as being red and white, but mine is, and that color combination is very useful in locating it.

One way to take advantage of the view invariance of color is to use a color histogram. Given a discrete color space defined by some color axes (e.g. red, green, blue), the color histogram is obtained by counting the number of times each color occurs in the image array. To illustrate, Figure 1 shows the three chromatic channels from a color camera together with a color histogram obtained from the images. In the example a set of opponent color axes are used, which are transformations of the red, green and blue axes [Ballard and Brown, 1982]. In a vision system that deals with large variations in lighting, the axes could be the basis functions for color constancy described in [Maloney and Wandell, 1986].

Histograms are invariant to translation and rotation about an axis perpendicular to the image plane, and change only slowly under change of angle of view, change in scale and occlusion. Figure 2 shows the variation in match value, which measures the fraction of pixels that fall into the same histogram buckets, between the model histogram and the image histograms as the view angle and distance are changed. Because histograms change slowly with view, a three-dimensional object can be adequately represented by a small number of histograms, corresponding to the object's *principal views* [Feldman, 1985]. Histograms are also efficient to compute. Generating a histogram from a 512 x 485 image takes about 40 milliseconds using MaxVideo image processing hardware, including the time needed to transfer the histogram to the host.

Both the object identification and object search implementations described in the following sections identify objects by matching image histograms to model histograms.

4 Object Identification

This section presents a method of matching image histograms to model histograms so as to efficiently recognize an object that appears in a real-world scene. Because the model database may be large, we can only afford a highly restricted amount of processing per model, but at the same time we must be able to overcome the problems that hinder recognition, most

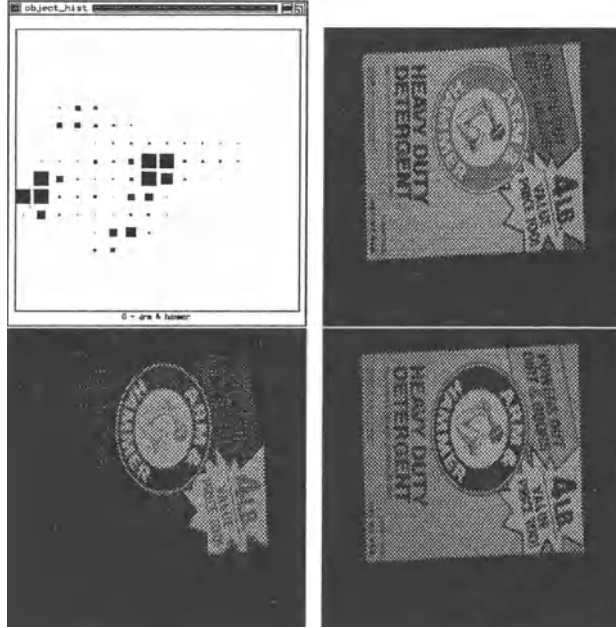


Figure 1: Clockwise from upper right: Red, green, and blue bands of “Arm & Hammer” image. The main body is yellow, the circle containing the hammer is red, the stripe at the top is green, and the lettering and hammer are blue and white. Upper left: Two-dimensional opponent color histogram of “Arm & Hammer” image, 16 buckets along each axis. Red-Green axis runs vertically, green at the top, red at the bottom. Blue-Yellow axis runs horizontally, yellow at the left, blue at the right. The yellow (far left) peak and black background (the center) peaks are the largest, and red and green peaks, as well as a small blue peak, are present (from [Wixson and Ballard 1989]). Three-dimensional opponent color histograms, used in the object identification section of this paper, have a third White-Black axis.

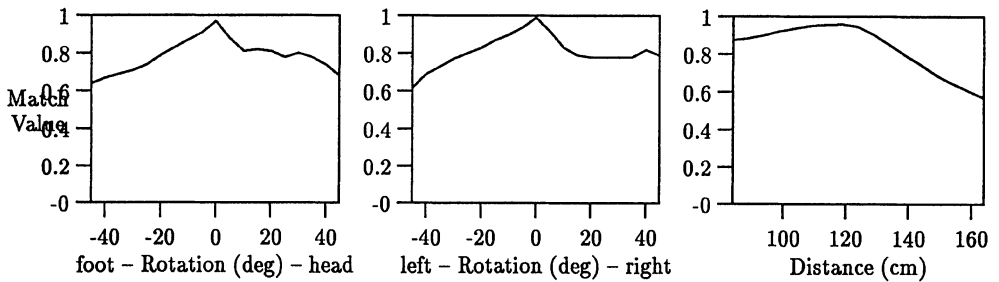


Figure 2: Variation of the Histogram Intersection match value as the camera is moved with respect to a Snoopy doll. In the Distance graph the model image was taken at a distance of 124 cm. The match value changes slowly with changes in angle and distance.

importantly

- distractions in the background of the object,
- viewing the object from a variety of viewpoints,
- occlusion,
- varying lighting conditions.

The matching method proposed here, called *Histogram Intersection*, is robust to the first three problems; the last is left to a color constancy module that operates on the input prior to the histogram stage. It is also extremely efficient and easy to implement. Two 16x16x8 histograms can be matched in 2 milliseconds on a SUN Sparcstation 1 (a 12 MIP RISC machine).

4.1 Description

Given a pair of histograms, I and M , each containing n buckets, the intersection of the histograms is defined to be

$$\sum_{j=1}^n \min(I_j, M_j).$$

The result of the intersection of a model histogram with an image histogram is the number of pixels from the model that have corresponding pixels of the same color in the image. To obtain a fractional match value between 0 and 1 the intersection is normalized by the number of pixels in the model histogram. The match value is then

$$\frac{\sum_{j=1}^n \min(I_j, M_j)}{\sum_{j=1}^n M_j}.$$

The Histogram Intersection match value is not reduced by distracting pixels in the background. This is the desired behavior since complete segmentation of the object from the background cannot be guaranteed. Segmentation is still a topic of active research, but the indications from the large amount of research done in the past are that segmentation is likely to continue to be a difficult problem that is computationally intensive and subject to failure

in many situations. The histogram intersection match value is only added to by a pixel in the background if

- the pixel has the same color as one of the colors in the model, and
- the number of pixels of that color in the object is less than the number of pixels of that color in the model.

Figure 2 shows the dependence of Histogram Intersection on viewpoint and occlusion. There are a number of ways of determining the approximate depth of an object, from laser or sonar range finders, disparity, focus or touching the object with a sensor. The depth value combined with the known size of the object can be used to scale the model histogram. Alternatively, if it is possible to segment the object from the background and it is not significantly occluded, the image histogram can be scaled to be the same size as the model histogram.

4.2 Representing a Large Database

Both theoretical and experimental examinations of histogram intersection demonstrate that the technique is suitable for indexing into a large database.

If objects are uniformly distributed in color space and all histograms are of the same (maximum) size, then the number of models that can be represented is at least

$$\frac{1}{(2\delta)^{n-1}}$$

where $1 - \delta$ is the minimum match value allowed and n is the number of buckets in a histogram [Swain and Ballard, 1991]. For any δ significantly smaller than $\frac{1}{2}$ this number is huge for size of n used in the implementation. For instance, if $\delta = 0.48$ and $n = 512$ (the number of buckets in an $8 \times 8 \times 8$ histogram) then the number that can be stored is about 10^9 . Smaller histograms match all the histograms of which they are a subset and therefore reduce the total number of objects that can be distinguished.

Experiments support the hypothesis that a large number of objects can be represented. For the 66 object database shown in Figures 3-5, the correct model is the best match 90%

of the time and is always one of the top two matches. Other, more expensive, matching techniques can be used to verify which of the top scoring models is the correct one, so it is not crucial that the correct model is always the best match. Notice that the histograms are not all the same size and that the objects are not uniformly distributed throughout color space (for instance, white is much more predominant than in a uniform distribution).

4.3 Efficient Indexing into a Large Database

Histogram Intersection is efficient compared to most recognition schemes. Nevertheless, for large databases the linear dependence of the recognition scheme on database size will add up. Parallel processing is one way of attacking this problem, since the match over different models is easily parallelizable. Another way, which reduces the recognition complexity to constant time for a broad range of databases, is shown below. In this scheme, called *Incremental Intersection*, only the largest buckets from the image and model histograms are compared, and a partial histogram intersection value is computed. The computation is incremental, so that the algorithm can be interrupted at any time with good results. This last feature could prove to be extremely important in a system that interacts with the real world, in which the times that actions are taken are often dictated by outside events.

Incremental Intersection is split into two phases, an off-line phase in which the data structure representing the database is generated and the on-line matching phase. In the off-line phase:

1. Assign to each bin in each model histogram a key which is (the number of pixels in the bin)/(total number of pixels in all bins)
2. Sort the bins by key and split into T groups. T is a parameter that adjusts the grain at which decisions are made as to which model bins to compare (see below).
3. Associate each group with a table indexed by color histogram bin. The table entries are linked lists pointing to all the histogram bins in the group with that index.

In the on-line phase:

1. Sort the image histogram bins by size.



Figure 3: Model indexing experiment based on color cues (continued in the next two figures). Each of the sixty-six models shown here is represented by its color histogram.



Figure 4: The unknown objects. Each is identified with the database color histogram that best matches its own color histogram.

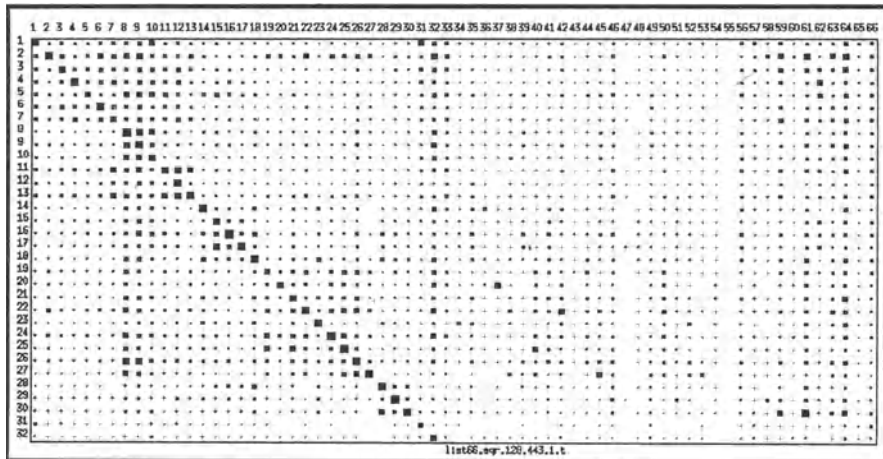


Figure 5: The results of matching all combinations of image and database histograms displayed pictorially where the size of the squares are proportional to match values. The dominance of the diagonal values shows that the correct match is almost always selected. Twenty nine of thirty-two matches are correct; in three cases the correct model received second highest score. Models are along the horizontal axis; unknown objects along the vertical axis.

Recognition Times (milliseconds)			
	Database Size		
	19	37	70
Histogram Intersection	38	73	150
Incremental Intersection	15	15	15

Table 2: Recognition times as a function of database size for the standard algorithm Histogram Intersection and the fast indexing scheme Incremental Intersection ($B=10$, $T=30$). Timings were made on a SUN SPARCstation 1.

2. For the B largest image bins, starting with the largest, match the image bin to all the model bins with the same index and in a table whose maximum key is larger. If a new table is entered in this process, match all the larger image bins to the model bins in that table.

The efficiency of Histogram Intersection and Incremental Intersection is compared in Table 2. A complexity analysis shows that the time for Histogram Intersection depends linearly on the size of the histogram and the size of the database, i.e. the algorithm is $O(nm)$, where n is the size of the histogram and m is the number of models in the database. As long as a bucket only indexes one model in the computation, the complexity of Incremental Intersection is independent of the size of the database. This explains the constant run-times of Incremental Intersection shown in the table. Asymptotically, the number of models each bucket indexes is linearly related to the number of models in the database. Therefore the expected asymptotic complexity of Incremental Intersection is also linear in the number of models (with a small constant factor), that is, $O(n \log n + cm)$, where c is the number of image bins used for indexing into the database (see [Swain and Ballard, 1991] for more details).

4.4 Improving Discriminability using Saliency

Recognition accuracy can be enhanced by concentrating on cues that distinguish an object from others in the database if the objects are similar in many ways. As an example consider a group of white shirts with different colored logos on them. Simple Histogram Intersection is incapable of distinguishing them, as shown in Figure 6(a). However, if the colors that show up in the rest of the database and are expected in the background are ignored, Histogram

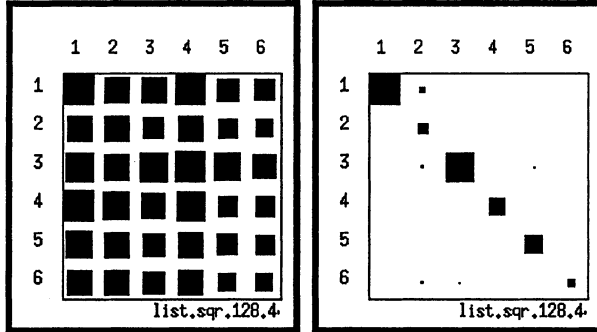


Figure 6: Experiment in distinguishing six white shirts with different colored logos. (a) Results of Histogram Intersection. (b) Results of modified Histogram Intersection, in which only colors unique to object model are matched. All the shirts are correctly identified.

Intersection can distinguish them all Figure 6(b). In this situation, the salient colors are defined to be those that are unique to an object and distinguish a particular object from the rest of the database.

The proposed indexing strategy is then: Consider all the colors equally to narrow the decision to a smaller number of similar objects and then apply a salience measure to differentiate among the objects.

The simple definition of salience described above is not the only useful one. For instance, combinations of histogram bins that are unique to an object could be used instead of the single bins used in the example.

5 Searching for Objects

Section 2 discussed possible uses for an object search mechanism. Let us now consider object search in more detail. Given a specification of the desired object, the goal of the object search mechanism is to produce a set of gazes (viewing positions and directions) that result in images that are likely to contain the object. It is important to realize that speed is an important component in this problem. If the mechanism is to be useful, it must recognize objects quickly. If it cannot find an object quickly, then it would be better to maintain a categorical map of the world such as that produced by image interpretation (despite the

problems with coordinate systems as described in Section 2) than to search for an object every time it is needed.

To achieve this goal, we have chosen to use a two-phase strategy. In the first phase, a fast object detection mechanism is used to detect the likely presence of the object in each image without performing pose estimation. The purpose of this phase is to prune out all but a small number of the possible gazes. In the second phase, more sophisticated object recognition algorithms are applied to the image produced by each gaze in order to confirm the object's presence and to compute its pose. This section describes an implementation of the first phase only. The object detection mechanism used in our experiments relied on matching image histograms to model histograms and was similar to the object identification algorithm discussed in the previous section, but modified to be more scale-invariant. (See [Wixson and Ballard, 1989] for more details.)

The search strategy discussed below is a generic search strategy suitable for finding an object that is somewhere in a room. Although clearly high-level knowledge such as "Cereal boxes are usually on the countertop" can be very useful for object search [Garvey, 1976], this knowledge will sometimes fail in unexpected situations. In these situations, some sort of default generic searcher must be used. It should be noted that it is in this case that pruning provided by the object detection mechanism is most important, due to the fact that we have no high-level knowledge to limit the search to certain portions of the room.

5.1 Sampling the Space of Possible Gazes

The most obvious generic strategy for object search is to position the robot in the center of the room and rotate the gaze 360 degrees around the vertical axis. To carry out object search experiments, we have mounted a Pulnix color camera on a Puma 761 robot arm [Brown, 1988]. The arm is mounted in the center of a $16' \times 24'$ room that contains cluttered scenes containing everyday objects. To search the room, the camera is positioned in the center of the room and rotated 360 degrees in increments of 15 degrees, with the field-of-view of the camera adjusted so that the spatial volumes seen by adjacent increments are spatially adjacent. This 360 degree rotation is executed for each of several pitch angles, so that the camera can examine the upper walls, the lower walls, and the floor. The object detector

mechanism, which produces a “confidence” that the object is in the scene, is run once at each of the gaze configurations.

After this sampling of the set of possible gazes is complete, the confidences are examined to determine which gazes produced “significant” confidences. A simple and effective mechanism for this is the criterion that for a confidence to be significant it must be at least one average deviation greater than the mean confidence for that object over all of the gazes evaluated. By expressing significance in terms of the distance from the mean, we avoid the use of thresholds that may vary with the surroundings (suppose we put new wallpaper or carpeting in the room) or with the specific object detection mechanism being used. The resulting set of significant gazes is then pruned further by eliminating gazes for which an adjacent gaze produced a larger confidence. For example, if the set contains gaze A = (orientation = 30° , pitch = 10° , confidence = .3) and gaze B = (orientation = $30 + x^\circ$, pitch = 10° , confidence = .7), where x is the rotation increment (usually 15°), then gaze A would be removed from the set. This is done so that when the desired object lies in the views of two gazes, only the better of those gazes is saved. The gazes remaining in the set after this pruning are those considered most likely to view the desired object.

5.2 Refining the Gazes

Since the above strategy samples discrete (although adjacent) volumes of space, it is possible that an object may appear only partially in any image. We have implemented a mechanism that, given an initial gaze, attempts to adjust the gaze to bring the object entirely into the picture. More specifically, the gaze is adjusted so as to maximize the confidence produced by the detection mechanism, *i.e.* to maximize the goodness of the object’s signature.

This maximization is done by gradient search in the space of possible gazes. Given a starting gaze, the arm translates the camera (relative to the camera’s initial gaze) by a small fixed amount up, down, left, right, forward and backwards. The detector’s confidence at each of these relative positions is saved, and the gaze is finally adjusted by the transformation that resulted in the largest increase in the confidence. This process is repeated until no move results in an increase in the confidence.

5.3 Summary

Our proposed search strategy can now be summarized. To search for a certain object,

1. Sample the space of possible gazes using a simple and fast object detection mechanism which produces a confidence value for each gaze.
2. Prune the gazes using their confidence values and relative locations, producing a set of significant gazes.
3. For each significant gaze (in order of decreasing confidence),
 - (a) Refine the gaze using the simple object detection mechanism;
 - (b) Invoke the sophisticated object recognition scheme to determine whether the object is really in the area covered by this gaze, and if so, to obtain the pose of the object.

Steps 1, 2, and 3a have been implemented. The following section presents the results of steps 1 and 2.

5.4 Performance

Generation of a 2-D opponent color histogram is performed at approximately $3\,512 \times 480$ frames/second by a Datacube MaxVideo real-time image processing system. Matching of the new histogram to the example histograms of the object being searched for is performed on a Sun 3/260. A typical search consists of rotating the camera around the vertical axis in increments of 15° at 3 different pitch angles, thereby requiring that 72 gazes be evaluated. Our system performs this evaluation in 3.5 minutes, taking just under 3 seconds to move to a new gaze, grab the histogram, and compute the match. This time is probably the best that can be done with our equipment in its current configuration, since moving to a new gaze takes one second, and a half-second delay is required between termination of the move and grabbing the frames for the histogram computation in order to allow the vibration of the camera to damp out sufficiently that the color will not suffer from motion blur.

The database of model histograms is usually acquired through an iterative process. Our method is to start with a small set of histograms of each object (usually two histograms, taken from two different positions), and run the search task. If an object is missed, we cover the background with black cloth, leaving the object in the same orientation, histogram the scene, and save the scene as an example of the object. By draping the background with black cloth, we eliminate the background signal from the histogram. Leaving the object in the same orientation ensures that the orientation and/or lighting effects that caused the object to be missed originally will be present in the new example histogram. The objects are then moved to different positions, the search task is executed again, and new histograms are learned if necessary. This process is continued until performance is deemed acceptable; the end result is that the database contains ~ 4 example histograms for each object.

Figure 7 shows the direction (but not the distance) of everyday multi-colored objects in a cluttered room in relation to the robot for a typical run of the search task. In addition to these objects, the room contains many other black, gray, or white objects such as tables, cabinets, TV monitors, bookshelves, and chalkboards. Figure 8 shows the gaze directions produced when the search strategy is executed for “Clorox” and “All”, detergent containers. In these figures the area of each circle is proportional to the confidence that the gaze in that direction includes the object. The numbers next to the circles reflect the ordering of the confidences in order of decreasing confidence; a circle with number 0 denotes the gaze that the system feels is most likely to contain the object. For both examples the proper gaze is gaze 0 (the highest-ranked gaze).

As described above, the manner in which we compile the test database results in the elimination of almost all false negatives. The false negatives encountered in practice are almost always due either to the presence of objects that massively obscure the signature of the object in the image, or to searching for an object that exhibits a large amount of specular reflection (for which it is difficult to compute stable example histograms). In typical runs, the correct gaze is almost never left out of the final set of gazes. Thus, this detection method meets the criterion that there be a low percentage of false negatives.

In theory, many possible images may give rise to the same histogram and hence to the same signature, resulting in a possibly large number of false positive matches. Our

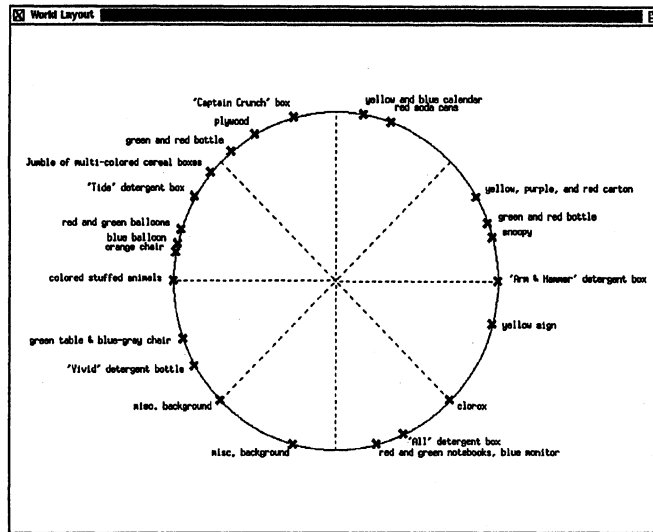


Figure 7: Top-view of the laboratory environment for a typical test run showing the direction (but not the distance) of each object with respect to the robot.

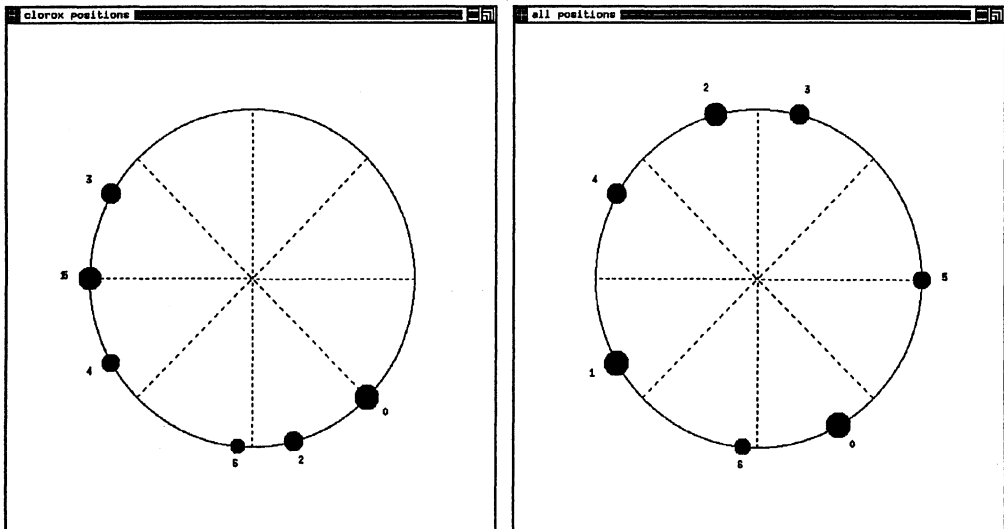


Figure 8: Gaze directions produced by the object search mechanism for the "Clorox" and "All" detergent boxes. Area of circle is proportional to the confidence in that gaze. Numbers next to circles reflect the ordering of the confidences in decreasing order.

experiments with the searcher and with making forced choice classifications from a database of images have shown, however, that this does not occur overwhelmingly often. The pruning strategy usually leaves only ~ 6 out of the 72 possible gazes for further processing¹; these are the gazes that have the highest confidence that they contain the desired object. This is more than a 90% reduction in the set of possible gazes. As stated above, one of these leftover gazes almost always contains the object, although this is not always the gaze that produces the maximum goodness. Thus, the false positives that are generated are not numerous enough to cause the correct gaze to be discarded by the pruning strategy.

5.5 Other Aspects of Object Search

There are many aspects of object search that have not been discussed in this paper. These include reasoning about occlusion and the use of high-level knowledge about common relationships between objects. For further discussion of these, see [Wixson, 1992; Wixson, 1990].

6 Conclusions

Object identification and search are two examples of tasks that are much simpler than image interpretation and produce much simpler spatial representations than image interpretation. Interestingly, however, they suffice in many situations where image interpretation would traditionally be applied. This lends credence to the animate vision hypothesis that many tasks do not require image interpretation in its totality. We have presented promising new techniques for identification and search that are based on a simple color histogram representation.

Acknowledgments

The “vision group” provided valuable advice and interesting discussions. This work was supported by NSF research grant DCR-8602958.

¹It is probably the case that up to eight leftover gazes are acceptable, since on average the object will be found in the fourth gaze and therefore a conventional recognition method that takes a minute to process a scene will take an average of four additional minutes (for a total of 7.5 minutes) to find the object.

References

- [Bajcsy *et al.*, 1989] Ruzena Bajcsy, Sang Wook Lee, and Ales Leonardis, "Image Segmentation with Detection of Highlights and Inter-Reflections Using Color," In *Image Understanding and Machine Vision, 1989 Technical Digest Series, Vol. 14*, pages 16–19. Optical Society of America, June 1989.
- [Ballard, 1989] Dana H. Ballard, "Reference Frames for Animate Vision," In *International Joint Conference on Artificial Intelligence*, pages 1635–1641, 1989.
- [Ballard and Brown, 1982] D.H. Ballard and C.M. Brown, *Computer Vision*, Prentice Hall, 1982.
- [Biederman, 1985] I. Biederman, "Human Image Understanding: Recent Research and a Theory," *Computer Vision, Graphics and Image Processing*, 32(1):29–73, 1985.
- [Brooks, 1987] Rodney A. Brooks, "Visual Map Making for a Mobile Robot," In Martin A. Fischler and Oscar Firschein, editors, *Readings in Computer Vision*, pages 438–443. Morgan Kaufmann, 1987.
- [Brown, 1988] Christopher M. Brown, "The Rochester Robot," Technical Report 257, Department of Computer Science, University of Rochester, 1988.
- [Feldman and Yakimovsky, 1974] J. A. Feldman and Yoram Yakimovsky, "Decision Theory and Artificial Intelligence: I. A Semantics-Based Region Analyzer," *Artificial Intelligence*, 5:349–371, 1974.
- [Feldman, 1985] Jerome A. Feldman, "Four Frames Suffice: A Provisional Model of Vision and Space," *The Behavioral and Brain Sciences*, 8:265–289, 1985.
- [Forsyth, 1988] D. A. Forsyth, "A Novel Approach to Colour Constancy," In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9–18, 1988.
- [Garvey, 1976] Thomas D. Garvey, "Perceptual Strategies for Purposive Vision," Technical Note 117, SRI International, 1976.
- [Healey and Binford, 1987] Glenn Healey and Thomas O. Binford, "The Role and Use of Color in a General Vision System," In *Proceedings of the DARPA IUS Workshop*, pages 599–613, 1987.
- [Hurlbert and Poggio, 1987] Anya C. Hurlbert and Tomaso A. Poggio, "Learning a Color Algorithm from Examples," In *Neural Information Processing Systems*, pages 622–631, 1987.
- [Klinker *et al.*, 1988] Gudrun J. Klinker, Steven A. Shafer, and Takeo Kanade, "The Measurement of Highlights in Color Images," *International Journal of Computer Vision*, 2:7–32, 1988.

- [Maloney and Wandell, 1986] Laurence T. Maloney and Brian A. Wandell, "Color Constancy: A Method for Recovering Surface Spectral Reflectance," *Journal of the Optical Society of America A (JOSA-A)*, 3(1):29-33, January 1986.
- [Maunsell and Newsome, 1987] John H. R. Maunsell and William T. Newsome, "Visual Processing in Monkey Extrastriate Cortex," *Annual Review of Neuroscience*, 10:363-401, 1987.
- [Mishkin and Appenzeller, 1987] Mortimer Mishkin and Tim Appenzeller, "The Anatomy of Memory," *Scientific American*, 256(6):80-89, June 1987.
- [Ohlander *et al.*, 1978] R. Ohlander, K. Price, and D.R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," *Computer Graphics and Image Processing*, 8:313-333, December 1978.
- [Rubner and Schulten, 1989] J. Rubner and K. Schulten, "A Regularized Approach to Color Constancy," *Biological Cybernetics*, 61:29-36, 1989.
- [Swain and Ballard, 1991] Michael J. Swain and Dana H. Ballard, "Color Indexing," *International Journal of Computer Vision*, 7:11-32, 1991.
- [Wixson, 1990] Lambert E. Wixson, "Autonomous Acquisition and Utilization of High-Level Spatial Knowledge to Search for an Object," Technical Report 338, University of Rochester Computer Science Department, April 1990.
- [Wixson, 1992] Lambert E. Wixson, "Exploiting World Structure to Efficiently Search for Objects," Technical Report 434, University of Rochester Computer Science Department, July 1992.
- [Wixson and Ballard, 1989] Lambert E. Wixson and Dana H. Ballard, "Real-Time Detection of Multi-Colored Objects," In *SPIE Sensor Fusion II: Human and Machine Strategies*, volume 1198, November 1989.

A MODEL OF HUMAN FEATURE DETECTION
BASED ON MATCHED FILTERS

M.C. Morrone and D.C. Burr

Istituto di Neurofisiologia del CNR,
Via S. Zeno 51,
Pisa, Italy

It is generally accepted that edge and line detection is an important stage of any visual system, biological or artificial. Many algorithms have been developed, either to simulate how humans may detect lines and edges, or as a stage in artificial image processing (see Hildreth, 1985). Most algorithms convolve the input image with operators of limited bandwidth, and search either for zero-crossings or peaks in the output.

There are several limitations inherent to all current algorithms. Convolution by an operator of limited bandwidth introduces ripples into the waveform, which can lead to spurious peaks and zero crossings, and hence false marking of features. A further limitation is that an operator designed to detect edges will also tend to mark lines (at an inappropriate position), and vice versa. For example, after convolution with a Difference-of-Gaussian operator, a line will cause a major peak corresponding to the centre of the line (the real feature), two strong zero-crossings on either side (whose distance from the peak depends on the spatial scale of the operator) and two weaker peaks further out from the zero-crossings. The spurious zero-crossings and peaks can lead to false marking of edges and lines. To minimize false positives attempts have been made to optimize the bandwidth and shape of operators, trading off precision of localization with signal detection efficiency (Canny, 1983). Other schemes

to minimize false positives include thresholding, requiring correspondence across spatial scales (Marr and Hildreth, 1980; Marr, 1982; Yuille and Poggio, 1985), using interpretation rules (Watt and Morgan, 1985), and resynthesizing images convolved with different types of operators (Canny, 1983). None of these schemes is perfect, and all face severe problems under certain features, such as adjacent edges of the same polarity, and combinations of edges and lines at the same point.

We have recently proposed a feature detection model for one-dimensional images, in which lines and edges are detected simultaneously by the same algorithm (Morrone and Burr, 1988; Morrone and Owens, 1987; Burr and Morrone, 1990). The algorithm has two stages, one linear and one non-linear. At the linear stage the image is convolved separately by two sets of operators with identical Fourier amplitude spectra, but with phase spectra which differ by $\pi/2$ (i.e. they are related by the Hilbert transform, and hence orthogonal functions in L^2). One of these operators is an even-symmetric function, designed to respond best to lines, the other odd-symmetric, designed to respond best to edges. After the convolutions, the output is squared separately and summed, to give the square of what we term the "Local energy" profile (following Adelson and Bergen, 1985). We have shown that for a wide range of patterns, local peaks in the energy function occur at points where visually salient features are perceived by human observers. These are the only peaks in the function; the spurious ripples introduced by the convolution process are annulled when the squared filtered images are summed. Thus to detect and locate features, it is sufficient to search for local maxima in the energy function, without the need for thresholding or use of interpretation rules to distinguish real features from false ones.

One of the main properties of this model is that both edges and lines cause peaks in local energy (unlike standard algorithms), so combinations of edges and lines (that can

occur naturally in shadows and conditions of oblique lighting) do not counteract each other to annul features. To determine whether the feature is an edge or a line or both, it is necessary to return to the linear stage of the algorithm, and evaluate the amplitude of the convolved images, at the points of local energy peaks. Non-zero amplitude in the even-symmetric convolution implies a line; non-zero amplitude in the odd-symmetric convolution implies an edge. If both are non-zero, the feature is a combination of line and edge.

The energy model has been shown to predict various perceptual phenomena in one-dimensional images, such as Mach bands (Morrone et al. 1986; Ross, Morrone and Burr, 1989), the Chevreul illusion (Morrone and Owens, 1987) and the Craik-Cornsweet illusion (Burr, 1987; Burr and Morrone, 1990), both quantitatively and qualitatively. It also predicts the perceived positions of complex combinations of lines and edges (Morrone and Burr, 1988), where other current models perform poorly.

The theoretical and mathematical justification for the energy model is given in previous papers (Morrone and Burr, 1988; Morrone and Owens, 1987), that show that local maxima of the energy function occur at points on the waveform where the Fourier harmonics come into phase. This suggests that maximal similarity in Fourier arrival phase (or argument) may be a useful definition of visual features. For edges, the average arrival phase is $\pi/2$, and for lines it is zero. This definition holds both for net edges and lines (step and delta functions) and for other features perceived as lines or edges (such as the illusory Mach bands, perceived at points where luminance ramps change slope). For all the waveforms we have tested to date, "arrival phase congruency" has proven to be a satisfactory definition of features, which helps explain the importance of phase in visual signals (Openheim and Lim, 1981).

The purpose of this paper is to extend the model to two-dimensional images, using operators like those of the one-dimensional model, with identical amplitude spectra but in quadrature phase. This constraint in two-dimensional space automatically introduces an orientation bias in the operators, so the image must be analysed by several operators of different preferred orientations. As with the one-dimensional model, we chose the parameters of the two-dimensional model to parallel as closely as possible those of the human visual system.

IMPLEMENTATION AND RESULTS

The proposed feature detection algorithm is similar to that developed for one-dimensional images. The local energy function is calculated by convolving the image with orthogonal operators of equal amplitude spectrum, and by then taking the pythagorean sum of the two separate outputs. Features are marked by the peaks of local energy, separately at four different orientations and three spatial scales. The feature maps are summed over orientation, to give a scale-by-scale description of the image, and can be summed over scales to give a global feature map.

Oriented matched filters

The first stage of the model is a linear convolution with two sets of masks, one odd-symmetric about a given plane, the other even-symmetric. The parameters for the masks were chosen from psychophysical and electrophysiological data from human and other primate visual systems. Both operators were selective to both orientation and spatial frequency. The amplitude spectra of the masks was given by equation 1, an equation that fits a range of psychophysical and electrophysiological data (e.g. Anderson and Burr, 1985, 1987; Hubel and Wiesel, 1977).

$$\begin{aligned}
 1) \quad & r(u,v) = g(u)f(v) \\
 & g(u) = \exp(-\ln^2(|u|/P)/2\sigma_u^2) \\
 & f(v) = \exp(-v^2/2\sigma_v^2)
 \end{aligned}$$

u and v are the frequency co-ordinates oriented along the preferred and non-preferred orientations respectively. $g(u)$, the spatial frequency tuning along the preferred orientation, is a Gaussian function on a logarithmic scale with a full band-width at half height of 2 octaves (given by $\sigma_u = 0.53 \ln$ units). Peak frequency p was 0.18 c/pixel. $f(v)$, the spatial frequency selectivity orthogonal to the preferred orientation, is Gaussian on the linear axis, with $\sigma_v = 0.11$ c/pixel. The orientation full bandwidth of $r(u,v)$ at half height is 70° .

Each image was analysed with pairs of masks at four orientations (0° , -45° , $+45^\circ$ and 90°) and three spatial scales (two octaves apart). Given the bandwidth of the operators, all useful spatial frequencies at all orientations were conserved at the linear stage, without excessive selective attenuation. The convolution masks were obtained by inverse Fourier transform of the amplitude spectra of equation 1, assuming a constant phase spectrum of either 0 or $\text{sign}(u)\pi/2$ (where u is the rotated coordinate at the preferred orientation). The phase spectrum is consistent with experimental data (Field and Nachmias, 1984; Burr, Morrone and Spinelli, 1989). Thus for a given orientation, the even- and odd-symmetric masks have identical amplitude spectra, and are related via the Hilbert transform. All oriented masks were 31×31 pixels and had the same preferred spatial frequency (at the preferred orientation). Analysis at different scales was achieved by the Gaussian pyramid technique, described in the next section.

Figure 1 reproduces contour plots of four examples of the masks. Figures A and B are masks oriented at 0° (with respect to the vertical); figure A is even-symmetric around the vertical plane, and figure B is odd-symmetric. Figures C and D are the even- and odd-symmetric masks at -45° orientation. Note that the masks at different orientations are not exactly identical, due to the grid-like under-sampling.

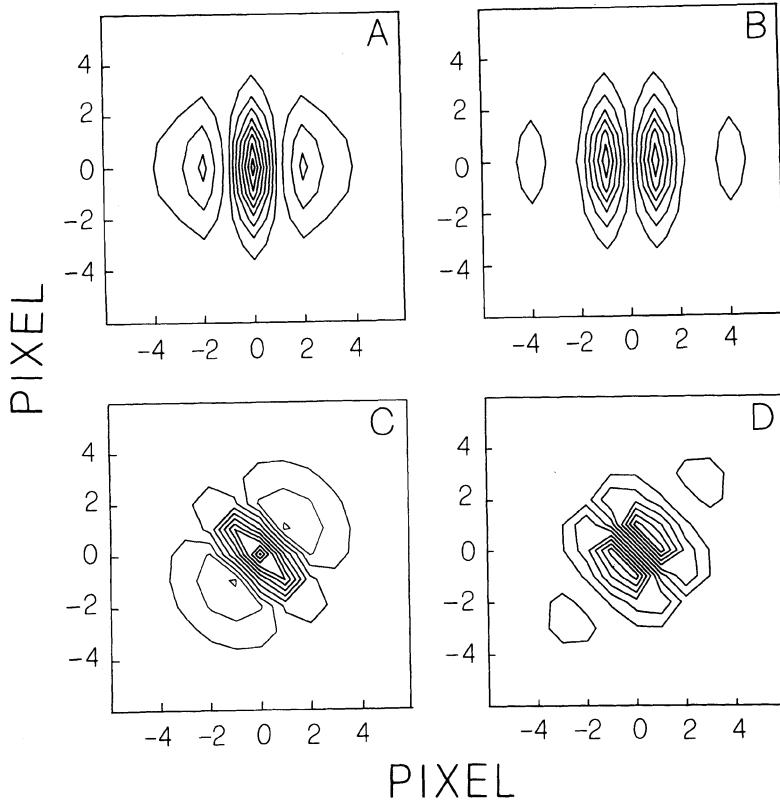


FIGURE 1

Contour plots of four of the eight masks used for convolution. A and B are orientated vertically (0°), C and D at -45° . The masks were calculated by inverse Fourier transform of the amplitude spectra given in equation 1, assuming a constant phase spectrum (see text). Masks A and C are even-symmetric about the preferred orientation, with a central positive region flanked by adjacent negative regions. Masks B and D are odd-symmetric, comprising separable regions alternating in polarity. Although only 13×13 pixels are depicted, the masks spanned 31×31 pixels, to minimize truncation effects.

FIGURE 2

Contour plots of local energy functions in response to a bright dot. The upper function was obtained with vertically oriented masks, the lower with masks oriented at -45° . Features are marked at local maxima at the orientation of the steepest gradient. For both orientations, the only feature marked is at the centre of the blobs, which corresponds to the position of the dot.

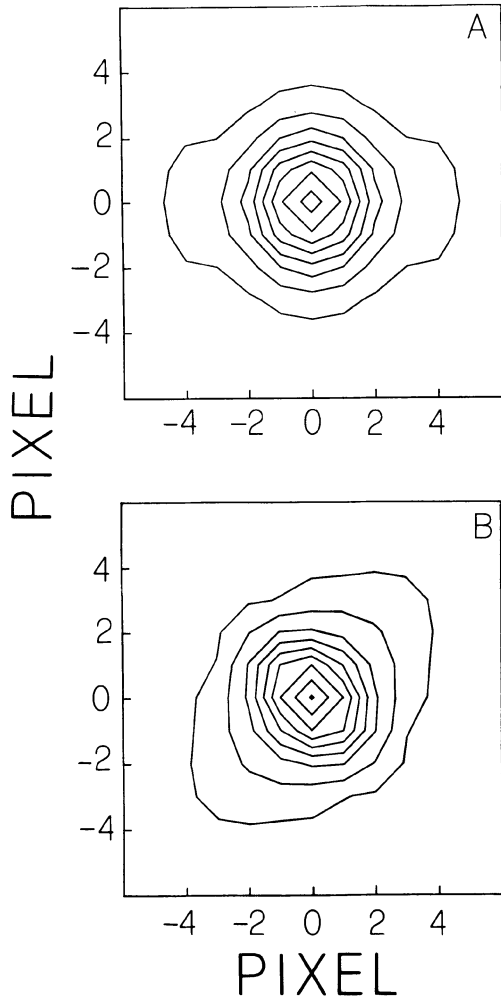


Figure 2 illustrates the application of the algorithm to a simple image, a small isolated dot. The dot was first convolved by the four pairs of orthogonal operators, to produce an output virtually identical to the profile masks themselves shown in figure 1. At each orientation, local energy functions were calculated by the taking the square-root of the sum of the squared output of the even- and odd-symmetric operators. Figure 2 shows examples of energy functions at orientations 0° (A) and -45° (B). The functions are smooth blobs slightly elongated along the non-preferred orientation of the masks. Features are marked by maxima in the local energy functions along the orientation where the

energy has maximum gradient (using the interpolation algorithm of Canny, 1983). This procedure minimizes overestimation of feature length, a common problem with elongated masks (see Canny). By inspection it is obvious that for both energy maps the only maxima along the direction of maximum energy gradient is the centre of the blobs, which correspond to the position of the input dot (without smear along the direction of the oriented masks). The final feature map is simply the sum of the features marked at the four orientations. No thresholding is necessary.

Spatial scales

As mentioned earlier, the operators are selective to spatial frequency as well as to orientation, with a bandwidth of about 2 octaves (like human visual detectors). To cover the range of useful image frequencies, the image was analysed at three scales, with each scale separated by two octaves (corresponding to the bandwidth of the operators). To minimize computer time, the analysis at the lower scales was achieved through reducing image size by a factor of four and using the same masks as for the higher scales. The images were reduced, and later expanded, by the Gaussian Pyramid technique of Burt and Adelson (1983). As well as reducing computer time, the pyramid compression technique maintains the mask sample grain at all scales, which is consistent with data from human visual detectors.

Figure 3 illustrates feature marking of an image at three scales. The upper figures show the original image, and the image reduced by factors of 4 and 16. The lower figures show the summed feature maps at each scale. As with the previous example, local energy was calculated separately for each orientation, and features marked at local maxima of energy along the direction of maximum energy gradient. The intensity at each marked feature corresponds to the amplitude of the local energy at that point. The four maps were summed across orientations at each spatial scale to produce the feature maps of figure 3.

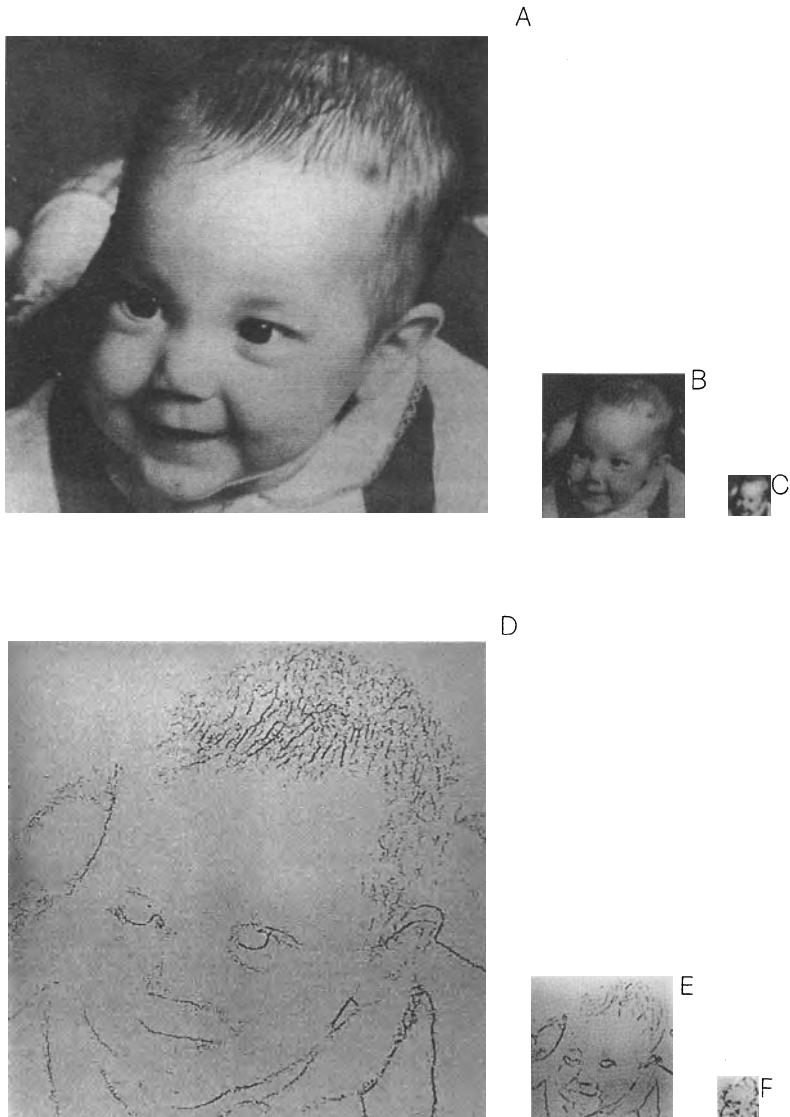


FIGURE 3

Application of the algorithm to a 513×513 pixel image at three spatial scales. The original image (A) was reduced using the Gaussian pyramid reduction technique, by a factor of four (B) and again by another factor of four (C). For each of the three images, local energy was calculated at four orientations and features marked at local maxima along the locally steepest energy gradient. The strength of the feature was given by the amplitude of the energy functions at those points. The lower images (D, E & F) show the feature maps at each scale, summed over orientation.

At each scale, the model marked the features of interest reasonably well, without need for thresholding or other noise rejection techniques. All types of features -- borders, lines, shading, speckle, etc. -- were marked with equal precision. To define the nature of each marked feature, we return to the linear stage of the model and consider the relative amplitudes of the even- and odd-symmetric operators. Positive amplitude in the even-symmetric convolution signals a bright line, and negative amplitude a dark line. Positive or negative amplitude in the oriented even- symmetric convolution signals positive or negative going edges. The operation at the highest scale provided the best sketch of the image, but the maps at the lower scales are also consistent with the outline of the baby face.

For many purposes, the feature maps at each scale would be a sufficient description of the image. It may be useful, however, to combine the information of the three scales to obtain a total feature map. We suggest that this could be done by expanding the maps at the lower scales (using the Burt and Adelson technique) to the size of the original image, and adding the three maps. During the expansion the amplitude is not scaled, so that the volume of the marked features remains the same as for the condensed image (but the amplitude is reduced). This has the effect of privileging information from the higher spatial scales. The technique is biologically reasonable, as it is known that the accuracy of position judgments scales with spatial frequency content of the image (Klein and Tyler, 1981).

An example of a combined feature map is shown in figure 4F. This image is sum of the maps from figure 3, after expanding the lower scales. Because of the automatic privileging of the smaller scales, they make the greatest contribution to the map. However, other features of interest, such as the outline on the left of the face, are better marked

at the lower scales. Given that there is no thresholding or post processing, the map depicts reasonably well the main features of the image.

Visual Illusions

A strong test of a model of human vision is whether it "sees" illusory forms in the ways that humans do. Many researchers (e.g. Barlow, 1959; Marr, 1976) have suggested that the human visual system searches for features of interest, and encodes them in some form of feature map, which would be a useful and economic code of the image (a similar process is almost certainly necessary in computer vision). If perceptual processes rely exclusively on a feature map for interpretation and structuring of images, much original image information could be lost or inaccessible.

Figure 4A is an example of a well known illusion, resulting from image coarse quantization (Harmon and Julesz, 1973). Although the image contains sufficient information at low spatial frequencies for recognition (verified by blurring the image, or viewing from a distance), the image cannot be recognized under normal viewing conditions: it appears as a collection of jumbled blocks. The edges of the blocks dictate the perceptual organization of the image, so that the observer loses access to the low frequency information (see Morrone, Burr and Ross, 1983).

Figure 4D shows the summed feature map obtained from the local energy algorithm, following the same procedure as that used to mark the figures of the original image (4F). The markings follow the outlines of the blocks, and are in no way associated with the form of the figure. If perceptual organization were based on a map of this type, the image would appear as a jumble of blocks, as human observers perceive it.

After digital blurring (figure 4B), the blocked image becomes recognizable. The feature map of the blurred image (again with the same algorithm) is shown in figure 4E.

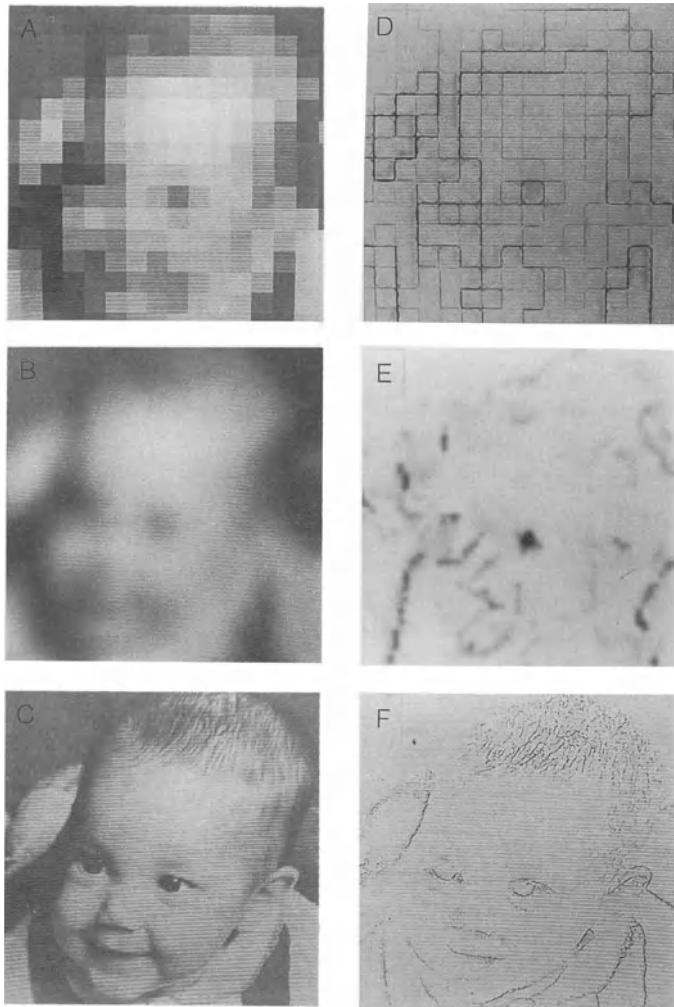


FIGURE 4

Application of the feature detection algorithm to coarse quantized images. The image of figure C (512×512) was blocked into a 16×16 matrix by averaging the luminance within each block. Figure B is a lowpass filtered version of figure A. On the right are the feature maps of the images, summed over orientation and three spatial scales (see text). The features of figure D follow the block outline of figure A, similar to the human perception of the figure. After blurring, however, the feature map is consistent with the outline of the original image. Figure F shows the summed feature map to the original image.

Although this map is not a very good description of the image, the markings do tend to follow the outline of the face, and to point to key features, such as the eyes. Recall that local energy marks all features, light and dark lines and positive and negative going edges. If the nature of the features were taken into account, a reasonable resynthesization may be possible. This aspect is currently being investigated.

Figure 5 shows another illusory figure, designed to demonstrate how congruence of arrival phase, and hence peaks in local energy, organize the perceptual appearance of images. Closely viewed, it appears as a chevron pointing left, while from a distance the chevron seems to point rightward. The equation for the figure is given in the caption. Each row comprises the sum of 64 cosine harmonics. The phase (but not the amplitude) of the harmonics changes systematically from 0 in the middle row to π in the upper and lower rows. This causes a systematic leftward shift in average luminance, resulting from the phase shift of the lower harmonics. There is also a group phase delay, such that the point where all harmonics have the same arrival phase shifts systematically rightward.

The lower figures show the features maps from the energy model, for the original figure (right) and for one blurred by digital lowpass filtering (left). Peaks in local energy occur at points of congruence of arrival phase (irrespective of the argument value at that point), so the feature map follows the lines of phase congruence and describes a chevron pointing leftward. Note that the features marked can be edges (arrival phase $\pi/2$ or $3\pi/2$), lines (arrival phase 0 or π), or combinations of edges and lines (all other phases). The local energy function marks all features equally well (whereas this pattern would clearly confuse most other algorithms). If perception were guided by a feature map similar to that given by peaks in local energy, the perception should be of a leftward pointing chevron, even though the average luminance moves in the other direction. When the pattern is blurred

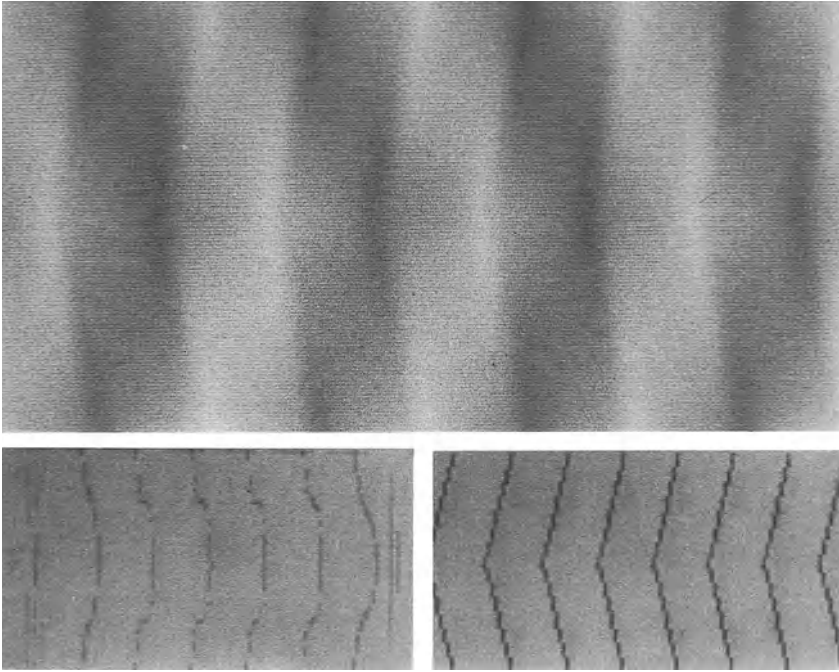


FIGURE 5

When viewed up close the upper figure appears as a chevron pointing leftwards; from a distance, or when blurred, it seems to point rightwards. The luminance profile $L(x,y)$ of the waveform is given by:

$$L(x,y) = L_0 + \frac{4a}{\pi} \sum_{k=1}^{\infty} \cos \left(2\pi \left[\left(\frac{y}{2T} - \frac{1}{2} \right) + k \left(\frac{1}{8} - \frac{y}{4T} \right) + k \frac{x}{T} \right] \right) \quad \text{for } y \leq 0$$

$$L(x,y) = L(x,-y) \quad \text{for } y > 0$$

For k an odd integer, $0 < x < 4T$ and $-T < y < T$. L_0 is the mean luminance, a amplitude and T the period. The lower figures show the feature maps, summed over four orientations and three scales. The right figure is the map of the original image, the left of the image blurred by digital filtering. Both maps predict the perceived structure, suggesting that the human visual system constructs maps similar to these.

(analogous to the optical blurring of distant viewing), the higher harmonics that reinforce the local energy peaks are attenuated, and the peaks in local energy tend to follow the organization of local luminance.

The final example of the action of the model is illustrated in figure 6. The input image is the upper left figure. The background pattern is made up of pyramidal shapes, created by multiplying a vertical with a horizontal triangle-wave luminance profile. The inner patterns are identical to the outer, on a smaller scale. There are two illusory effects with this pattern. Firstly, the perceived bright and dark crosses are illusory: the luminance profile is locally pyramidal with the apexes at the centre of the perceived crosses. The crosses are the two-dimensional equivalent of Mach bands (Mach, 1865). Another illusory effect is that the borders of the innermost pattern do not seem to be straight, but bulge inwards. If the pattern is viewed from a distance, the borders of all the inner patterns bulge in the same way.

The other images of figure 6 show the feature maps of the local energy model (for the middle scale only). In this case the maps do not depict total energy (as in the previous examples), but the contribution of the various types of features. The upper right photograph is the map of dark lines, the lower left bright lines and the lower right edges (of both polarities). The maps were produced by first marking all features at each orientation (by the method described previously), and then determining the amplitude of the feature (at each orientation) from the output of the linear convolution. The dark lines are given by the absolute value of negative values from the even-symmetric operator, the bright lines from positive values of the even-symmetric operator, and edges from the odd-symmetric operator (positive and negative).

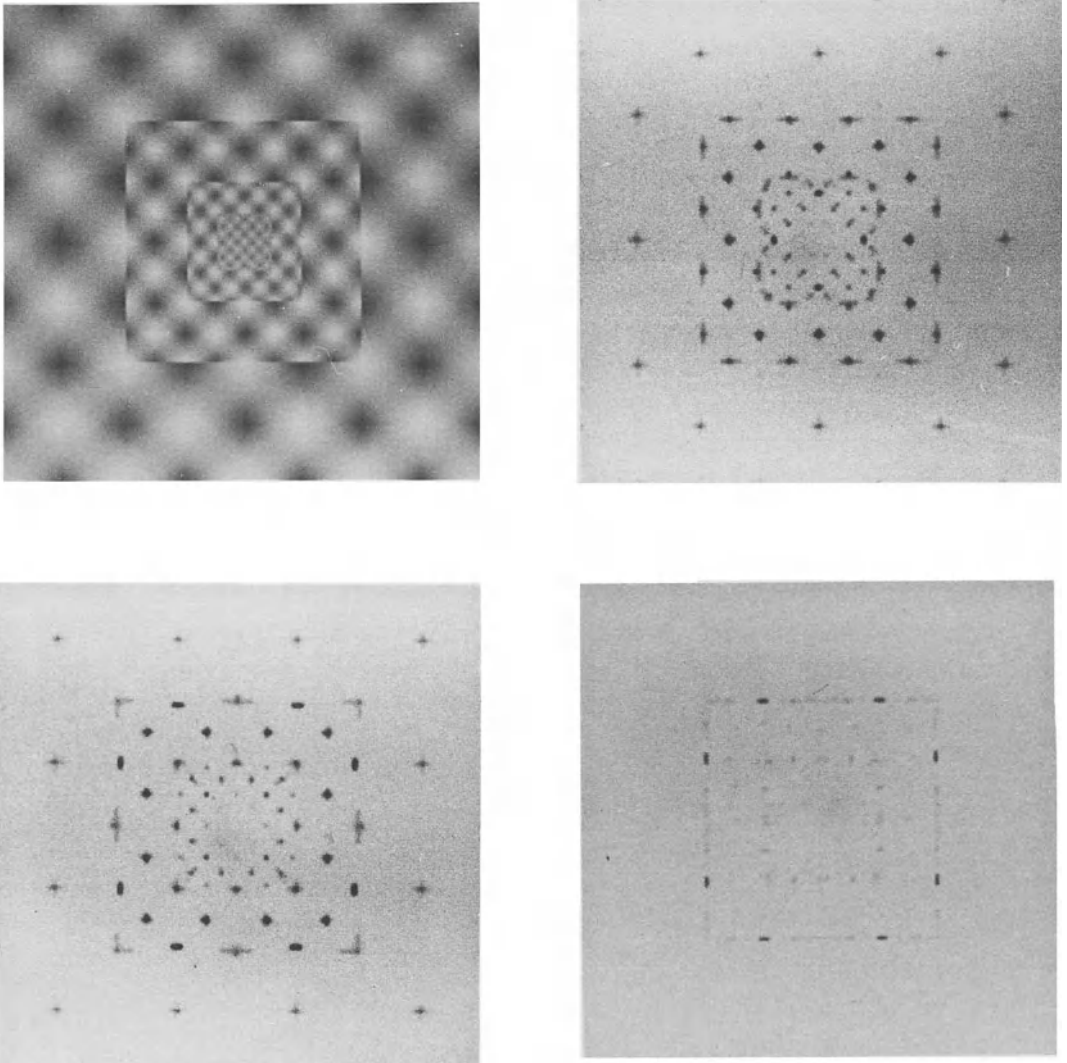


FIGURE 6

Examples of separate line and edge maps of a complex image. Figure A shows the image, B the map of marked dark lines, C the map of marked bright lines and D the map of edges (of both polarities). The outer part of the image can be defined as the product of a vertical and horizontal triangle-wave, with four periods per picture. The inner versions are identical, each scaled by a factor of two. The edge and line maps predict the illusions created by the stimulus, both the cross-like Mach bands, and the bowing of the inner borders with increased viewing distance.

The crosses are well marked by the model, either as bright or dark lines; they do not appear in the edge map. The edge map marks well the borders of the larger inserts (as straight lines), but only very weakly the innermost border. On the other hand, the inner border is marked as a dark line, curving inwards (as it appears to observers). Again the model predicted well the appearance of a complex pattern, and showed the importance of lines and edges in structuring the perceptual organization of visual images.

DISCUSSION

We have presented an algorithm of edge and line detection designed to mimic the human visual response to image features. The algorithm locates features by calculating the local energy function (defined as the pythagorian sum of the image convolved with operators related by Hilbert transform) and searching for local maxima. It then identifies the nature of the marked feature by evaluating the amplitudes of the two separate convolution outputs at the point where the feature was marked. The algorithm was tested with simple images, and images that cause illusory effects. In all cases the algorithm marked features at the points where observers see them, even when the perceived feature was not obvious from the luminance profile.

As the algorithm is primarily a model of human vision, we were guided by current knowledge of the mechanisms of human and animal visual systems, and chose the parameters of the algorithm from psychophysical and electrophysiological data. In the primary visual cortex of primates (and most other mammals), all neurones are selective to stimulus size and orientation (e.g. Maffei and Fiorentini, 1973; Hubel and Wiesel, 1977). There exist two classes of cells, one quasi-linear (simple cells) and the other clearly non-linear (complex cells)). Simple cells have limited spatial frequency response (1-2 octaves) with line-spread functions that tend to be even- or odd-symmetric (Kulikowski and Bishop, 1981). These

cells are grouped in the cortex in a way that could enable them to act as the matched filters of the linear stage of the algorithm (Pollen and Ronner, 1981). Complex cells also have a limited spatial frequency response, but exhibit a clear second-order non-linearity (Spitzer and Hochstein, 1985), consistent with the squaring and summing of local energy extraction.

Features were marked by local energy maxima only along the orientation of maximum energy gradient. A plausible biological mechanism for this non-maximum suppression may be "cross-orientation inhibition", observed electro-physiologically in man and other mammals (Morrone, Burr and Maffei, 1981; Morrone and Burr, 1986; Burr and Morrone, 1987). Cells of different orientation selectivity inhibit each other, so that the relative response of the preferred orientation is increased.

There is as yet no strong evidence that biological visual systems do calculate a function similar to local energy, or use it to mark features. However, the operations in the model are biologically plausible, and are appropriate for a limited bandwidth parallel computer, such as the human brain. The fact that the algorithm behaves quantitatively like humans (Morrone and Burr, 1988; Ross, Morrone and Burr, 1989; Burr and Morrone, 1990) and "sees" the same illusions that human observers do is encouraging.

Although designed primarily as a model of human vision, the algorithm may be a useful stage in computer visual systems as a precise and robust edge and line detection system. The algorithm has several potential advantages over current methods of feature detection. Both lines and edges, as well as other features with no local symmetry, are detected simultaneously. Each feature is marked uniquely, reducing the problem of false marking. It is relatively robust to noise, as local energy is maximum at points of congruence of arrival phase. Noise (by definition) has a random phase spectrum, and hence a low probability of arrival phase congruency at any

point. Chance aggregations of arrival phase will tend to occur at random isolated points, easy to remove by standard rejection techniques.

With the current implementation, the algorithm is expensive in computer time, requiring 24 separate convolutions of the input image, 8 of them on the full size image. However, with specialized hardware, and use of orientation pyramid techniques (Adelson and Simoncelli, 1987), this time could be reduced considerably, possibly to acceptable commercial levels. We are currently evaluating the performance of the algorithm (accuracy, noise immunity etc.) against standard algorithms.

ACKNOWLEDGEMENTS

This work was supported by grants from the Australian National Health and Medical Research Council, and the Italian Consiglio Nazionale delle Ricerche, Progetto Finalizzato Robotica (contract N. 89.00536.67).

REFERENCES

- Adelson, E.H. & Bergen, J.R. (1985) Spatio-temporal energy models for the perception of motion J. Opt. Soc. Am. A2 284-299.
- Adelson, E.H. & Simoncelli, E. (1987) Orthogonal pyramid transforms for image coding. Proc SPIE 845 50-58.
- Anderson, S.J. & Burr, D.C. (1985) Spatial and temporal selectivity of the human motion detection system Vision Res. 25 1147-115.
- Anderson, S.J. & Burr, D.C. (1987) Receptive field sizes of human motion detectors Vision Res. 27 621-635.
- Barlow, H.B. (1959) Possible principles underlying the transformations of sensory messages. In Sensory communication (Edited by W.A. Rosenblith). MIT Press, Mass, USA.
- Blakemore, C. & Campbell, F.W. (1969) On the existence of neurones in the visual system selectively sensitive to the orientation and size of retinal images. J. Physiol. (Lond.) 225 437-455.

- Burr, D.C. (1987) Implications of the Craik-O'Brien illusion for brightness perception. Vision Res. 27 1903-1913.
- Burr, D.C. & Morrone, M.C. (1987) Inhibitory interactions in the human visual system revealed in pattern visual evoked potentials. J. Physiol. (Lond.) 389 1-21.
- Burr, D.C. & Morrone, M.C. (1990) Edge detection in biological and artificial visual systems. In Vision: Coding and Efficiency (Edited by C. Blakemore). CUP, Cambridge.
- Burr, D.C., Morrone, M.C. & Spinelli, D. (1989) Evidence for edge and bar detectors in human vision. Vision Res. 29 419-431.
- Burt, P.J. & Adelson, E.H. (1983) The laplacian pyramid as a compact image code. IEEE Trans COM 31 532-540.
- Canny, J.F. (1983) Finding edges and lines in images. MIT AI Lab. Tech. Report 720.
- Field, D.J. & Nachmias, J. (1984) Phase reversal discrimination. Vision Res. 24 333-340.
- Harmon, L.D. & Julesz, B. (1973) Masking in visual recognition: effect of two-dimensional filtered noise. Science 180 1194-1197.
- Hildreth, E.C. (1985) Edge detection. MIT AI Lab memo. 835.
- Hubel, D.H. & Wiesel, T.N. (1977) Architecture of macaque monkey visual cortex. Proc. Roy. Soc. Lond. B198 1-59.
- Klein, S.A. & Tyler, C.W. (1981) Phase discrimination of single and compound gratings. Invest. Ophthal. Vis. Sci. S20 124.
- Kulikowski, J. J. & Bishop, P. O. (1981) Linear analysis of the responses of simple cells in the cat visual cortex. Exp. Brain Res. 44 386-400.
- Mach, E. (1865) Über die Wirkung der räumlichen Vertheilung des Lichtreizes auf die Netzhaut. I.S.-B. Akad. Wiss. Wien, math 54 303-322.
- Maffei, L. & Fiorentini, A. (1973) The visual cortex as a spatial frequency analyzer. Vision Res. 13 1255-1267.
- Marr, D. (1976) Early processing of visual information. Phil Trans. R. Soc. Lond. B275 485-526.
- Marr, D. (1982) Vision Freeman, San Francisco.
- Marr, D. & Hildreth, E. (1980) Theory of edge detection. Proc. R. Soc. Lond. B207 187-217.
- Morrone, M.C. & Burr, D.C. (1986) Evidence for the existence and development of visual inhibition in humans. Nature 321 235-237.
- Morrone, M.C. & Burr, D.C. (1988) Feature detection in human vision: a phase dependent energy model. Proc. R. Soc. (Lond) B235 221-245.

- Morrone, M.C., Burr, D.C. & Maffei, L. (1982) Functional significance of cross-orientational inhibition: part I Neurophysiology Proc. Roy. Soc. (London) B216 335-354.
- Morrone, M.C., Burr, D.C. & Ross, J. (1983) Added noise restores recognition of coarse quantised images. Nature 305 226-228.
- Morrone, M.C., Ross, J., Burr, D.C. & Owens, R. (1986) Mach bands depend on spatial phase. Nature 250-253.
- Morrone, M.C. & Owens, R. (1987) Feature detection from local energy. Pattern Rec. Letters 1 103-113.
- Openheim, A.V. & Lim, J.S. (1981) The importance of phase in signals. Proc. IEEE 69 529-541.
- Pollen, D.A. & Ronner, S.F. (1981) Phase relationships between adjacent simple cells in the visual cortex. Science 212 1409-141.
- Ross, J., Morrone, M.C. & Burr, D.C. (1989) The conditions for the appearance of Mach bands. Vision Res. 29 699-715.
- Spitzer, H. & Hochstein, S. (1985) A complex-cell receptive field model. J. Neurophysiol. 53 1266-1286.
- Watt, R.J. & Morgan, M.J. (1985) A theory of the primitive spatial code in human vision. Vision Res. 25 1661-167.
- Yuille, A.L. & Poggio, T. (1985) Fingerprint theorems for zero-crossings. J. opt. Soc. Am. A2 683-692.

Visualizing and Understanding Patterns of Brain Architecture

*Alan Rojer
Eric L. Schwartz*

Department of Computer Science
Courant Institute of Mathematical Sciences
715 Broadway
New York, N.Y. 10003

Computational Neuroscience Laboratories
New York University School of Medicine
Department of Psychiatry
550 First Avenue
New York, N.Y. 10016

ABSTRACT

We illustrate application of computer science to neuroscience at three levels: measuring, modeling, and understanding the computational function of the columnar pattern of ocular dominance in primate visual cortex. We review our methods for the quantitative reconstruction of the pattern of binocular input to the visual cortex of monkeys. We show that an oriented bandpass filter, applied to white noise, provides a simple parametric characterization of the observed pattern. We suggest a computational motivation for the columnar architecture as a “brain data structure” for a stereo vision algorithm based on the properties of a nonlinear filter, the cepstrum. This work illustrates some of the algorithmic difficulties and novel research problems encountered when computational approaches are used to visualize the patterns of neural architecture of the primate visual system.

Computational neuroscience: beyond rendering

One way of defining the term “computational neuroscience” is as the set of problem domains in which there is a nontrivial overlap of neuroscience and computer science (Schwartz 1990). “Non-trivial” is a loaded term. It was not so long ago that the use of computer data acquisition in microelectrode and EEG experimentation was sufficiently exotic to be considered a “computational” topic. But today, this type of application is routine. Real-time control of electrophysiological data acquisition is a standard skill; it has become a problem of neuroscience, and not of computational neuroscience.

Applications of computer graphics to anatomical reconstruction have similarly become routine, at least at the level of displaying a simple 3D model based on either voxel or polygon solids models. Most vendors of high-end graphics equipment provide prepackaged routines for the display of brain data. Given a set of serial sections of brain, either in images (voxel data structure) or wire-frame (polygon data structure) form, it is

routine to construct a hidden surface rendering on a graphics display. Such renderings of brains have become a staple of educational television and popular science publications. But these reconstructions do not address the deeper scientific issues of meaningful modeling of neural structures and, more importantly, of understanding why they are important for brain function.

We emphasize that the practice of computational neuroscience naturally divides into three levels: measurement and reconstruction, modeling, and computational significance. In the present paper, we will review recent work from our lab at each level. Specifically, we will apply computational methods to the elucidation of the structure and function of a particular architectural feature of the primate brain: the ocular dominance column (ODC) system, which comprises the first area of interaction of the projections from the left and right eyes to the visual cortex in primates. Since all the examples in this paper are drawn from this system, the diverse areas of expertise which are involved are somewhat unified. This work provides a fair sample of the scope of the research problems which characterize computational neuroscience.

1. Level 1: Measurement of the Macaque Ocular Dominance Column System

Binocular stereo vision is based on the brain's ability to detect and utilize small differences between a pair of projections of a single 3D scene, due to the displacement of the two eyes. In monkeys, information from separate eyes is not integrated until it reaches primary visual cortex (V1, also known as striate cortex and area 17). When the roughly 10^6 neural fibers (axons) from each eye enter V1, they segregate into a striking pattern of stripe-like domains, as shown in Figure 1. The ODC pattern is shared by many primate species, as well as other mammals (LeVay *et al.* 1975).

There are more than 20 functionally distinct areas of visual cortex in primates. And all of them are folded, twisted, and stacked up in a very uncongenial fashion. Depending on the anatomical and functional criteria, there are some 5-10 distinct layers in each cortical area. In monkey cortex, which is about 1 mm thick, the central layer of the cortex where the pattern is most prominent is several hundred microns thick. Sections from a serially sliced brain are shown in Figure 2. The gap between the anatomical data of Figure 2 and the large-scale pattern of Figure 1 illustrates the challenge in visualizing neural patterns.

Computational neuroanatomy: problem specification

The previous brief introduction presents the neuroanatomists dilemma. When a brain is cut in conventional serial sections (as in Figure 2), it is very difficult to visualize the kinds of patterns which exist in the *tangential* plane of the cortex. Patterns of interest are embedded in interior layers of cortex. Large sections of the cortex are obscured by the folds of sulci and gyri. The data is distributed in a series of cross-sections which obscure the tangential structure which is of interest. This leads us to formulate some essential problems of computational anatomy. Given a set of serial sections of cortex, we wish to:

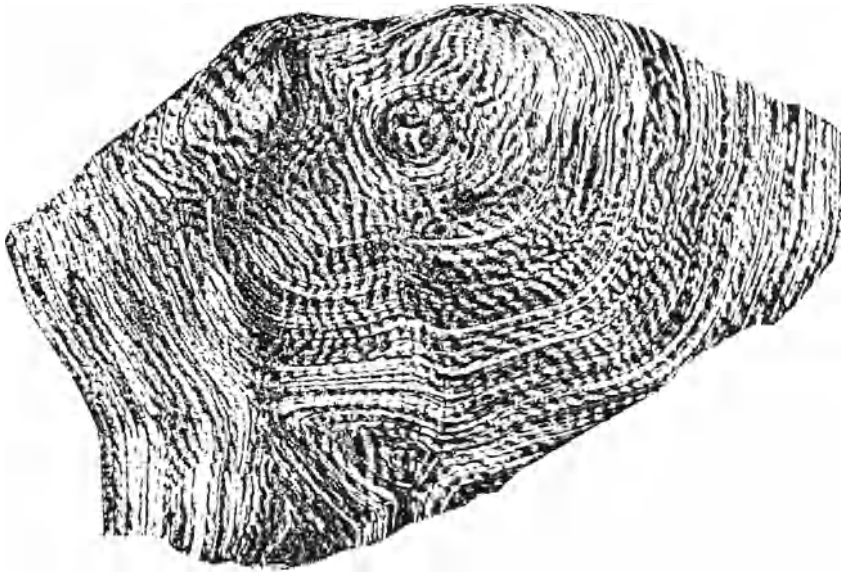


Figure 1 A computer-flattened brain peel at the level of layer IV of visual cortex of a macaque monkey, stained to reveal the pattern of ocular dominance columns (dark: left eye; light: right eye).

1. Construct a high-resolution solid model of the surface of the cortex.
2. Peel the layers apart and render them in gray-scale.
3. Construct joint voxel and polyhedral surface models.
4. Unfold the polyhedral cortex into a planar model
5. Texture map the 3D image pattern into 2D

We omit from this discussion the difficult problem of producing the pattern of anatomical “stain” which reflects the neural pattern of interest. This is a problem of neuroscience, not computational neuroscience. We will assume that we have obtained a “stain” which marks out the patterns of interest to us.

Construction of a solid (voxel) model of cortex

Given a set of raw serial sections, appropriately stained so that the ODC stripes are prominently marked (e.g. the left eye projection is dark and the right eye projection is light), the first problems encountered are digitizing, aligning, and conditioning the digitized sections. The serial sections comprise thin slices of brain tissue, stained and mounted on glass slides (see Figure 2). The sections are 40μ thick, and there are about four hundred sections for V1. We digitize our sections using a CCD camera which is

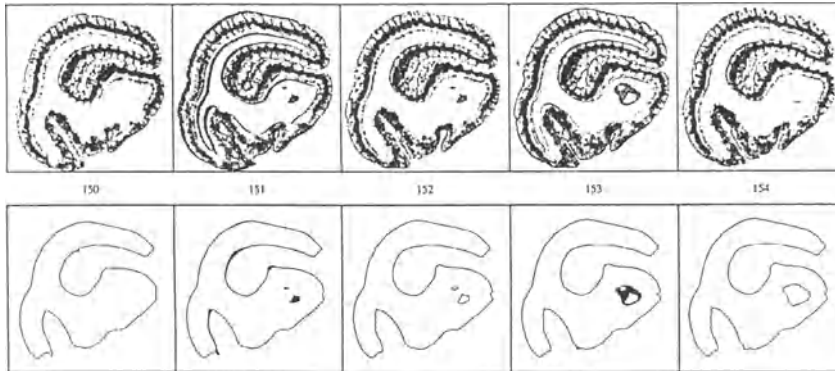


Figure 2 On top are five serial sections, cut in the coronal plane of the visual cortex of the monkey used in Figure 1 above. Even though the cortex is a simply connected shell, the complexity of folding of the surface leads to multiply-connected components in section. The evolution of a saddle point (dark figure eight in section 153) which is internal to the main component of cortex can be observed. On the bottom are the corresponding contours produced by intersecting a brain peel (see Figure 3) with a simulated cutting plane. These five contours correspond to the sections shown above, and lie roughly in layer IV of the cortex.

mounted under a conventional photographic enlarger. The slides are small enough to be inserted directly into the film carrier, and appropriate optics minifies the image onto the surface of a CCD chip, which is then digitized in NTSC video format. The CCD chip is mounted on a rotation and translation stage, which can be manipulated with great precision.

To reconstruct the solid brain, sections cut from the face of a block of brain must be precisely realigned. A certain amount of distortion affects each section. In principle, such distortion may be fixed by generalized image warping of the sections. Fortunately, we have had satisfactory results from similarity transformations (size, shift and rotation). We have not been forced to include general affine or polynomial warps in order to reconstruct the samples of V1 which we have processed,

It is a common practice to insert pins through a brain to be sectioned, and then to align adjacent sections up to these pin marks. To avoid damaging the tissue, we instead photograph the frozen brain block as it is being cut. This provides a record of the physical position of a serial section before it was cut, in the form of a "movie" of brain sections. The frames of this movie are digitally toggled with the images of the sections in the digitizer. The sections are moved physically to achieve alignment. This method works extremely well, allowing us to align a full set of serial sections with an estimated RMS error of about 150μ (Schwartz and Merker 1986b).

Once the sections are aligned and digitized, they undergo a series of conditioning steps.

1. *Thresholding* to isolate the gray matter of the cortex from the other brain structures in the digitized image.
2. *Histogram equalization* to stretch contrast and normalize the range of gray scale.
3. *Image Filtering* to repair damage due to blood vessels, microtome knife chatter, etc.
4. *Interactive Repair* to be sure that no topological defects exist in the images.

Figure 2 shows the final results of these processes.

Peeling the Brain

The steps so far described yield an implicit solid model of the brain in the form of several hundred images, each of a single serial section. Since these sections are aligned and conditioned, their union is a digital representation of the ODC pattern. But in this form the representation is of limited usefulness. As mentioned above, we need access to internal layers: we need to peel the brain. A microtome peels the brain by cutting it into thin sections, parallel to the knife blade. We wish to digitally cut the brain in curved sections which are parallel to the tangent plane of the brain. We now describe two algorithms to produce brain peels.

Intersection of a polyhedron with a voxel model

This algorithm is relatively trivial. Given a polyhedral model of an interior layer of a solid, the polyhedron can be intersected with the 3D voxel model to obtain a voxel surface. The polyhedral model may be produced by hand tracing of contours on individual images of brain sections, followed by conventional (interactive) 3D triangulation. One advantage of this method is that it can produce a surface at a known (i.e. physiological) location, subject only to the accuracy and knowledge of the hand tracing of contours. However, this method will only reliably produce a voxel surface within a few voxel thicknesses of the traced surface. As one moves further from this surface, the “peels” produced this way become unreliable, since they are following a relatively coarse approximation to the voxel surfaces. If one wishes to produce a full set of peels (e.g. 25 peels at 40 micron thickness, spanning the full thickness of cortex), this method is inadequate. It would require a great deal of laborious hand tracing and hand triangulation.

3D surface tracking with a shield

Excellent algorithms for 3D surface tracking are known (e.g. (Artzy *et al.* 1981). Surface tracking provides two surfaces: the inside and the outside of the 3D voxel model. This is problematic, because we wish to consider the inside and the outside of the voxel model as distinct surfaces, which is the case physiologically. It is thus necessary to construct a “shield” which will prevent a surface tracker from producing both the inside and outside surfaces of the solid shell as a single connected component.

Successive peeling can also introduce topological problems when applied to complex surfaces. A given surface may be perforated, thus changing the connectivity of the

residual surface. Small connected components (“dirt”) may escape image conditioning or may be produced in the process of peeling. These details must be carefully dealt with, as the process of peeling is subject to extreme topological instability: a single voxel spuriously connecting two surfaces can change the qualitative nature of the entire peel being produced.

Figure 3 shows a peel of a monkey brain produced by this algorithm incorporating the ODC pattern. Further details of our work in surface tracking and brain peeling may be found in (Frederick and Schwartz 1990).

Joint Voxel and Polyhedral Models: Automatic Triangulation

The brain peels which we produce contain all of the relevant data for ODC measurements. Unfortunately, the peel reflects all the irregularities of curvature and folding of the cortical surface. The problem of unfolding and flattening the cortical surface will be addressed in the next section; we note here that a later requirement for extensive numerical calculations (brain flattening) necessitates a more compact model of the peel than that afforded by voxels. We therefore require a conversion from the high-resolution, storage-intensive voxel representation of the peel to a piecewise linear (polyhedral) representation which captures the geometry of the peel with a much smaller storage burden. We thus need to *triangulate* the voxel surface.

There have been many algorithms described in the literature for the purpose of constructing polyhedral surface models, especially from serial sections. Early version of these algorithms (e.g. (Fuchs *et al.* 1977), could not deal with surfaces which had multiple contours per slice. Recently, several algorithms have been described (Anjyo *et al.* 1987) Boissonnat 1988 ; which are promising for use with complex surfaces, but they rely on heuristics which might prove troublesome with highly complex surfaces, such as cortex.

The difficulties of automatic triangulation can be traced to a single problem: when a three dimensional surface is represented by the union of its contours, there can be topological changes in the connectivity of the contours which are caused by the existence of critical points in the surface. This problem is illustrated with a sectioned torus (Figure 4). A torus in general position has four critical points, corresponding to places where the tangent plane to the torus is horizontal. These can be viewed locally as a maximum, a minimum, and two saddle points. The sectional representation of the torus undergoes topological changes when these critical points are traversed.

As the the height of the sectioning plane increase, the first nontrivial section comprises a lone point (locally a minimum). Then the cross section becomes a simple loop. At the next critical point (a saddle), the section becomes a figure eight; after the saddle point there are two simple loops. The third critical point is also a saddle (with a figure-eight section), after which the contours merge back into a single loop. The final critical point is the sole point in the last nontrivial section.

The existence of critical points, and hence changes of topology between adjacent sections, provides a major challenge to triangulation algorithms. In order to deal with

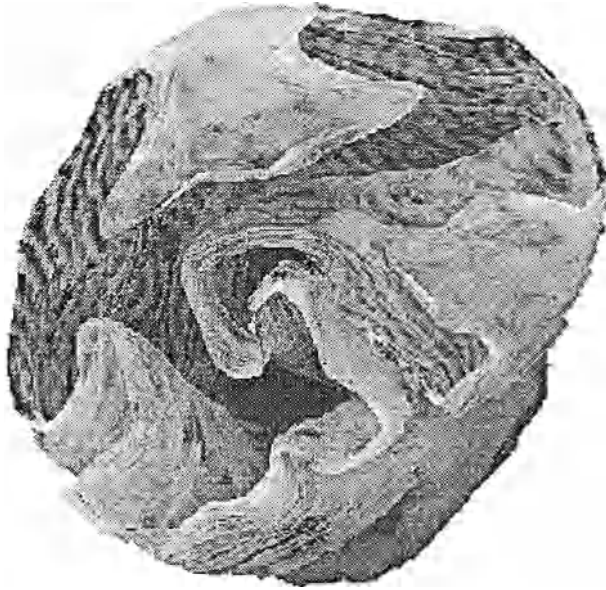


Figure 3 A brain peel of monkey brain, stained to reveal the ocular dominance column pattern. This shell is about 40 thick, and corresponds roughly to layer IV of V1 (visual cortex). Shown here is the majority of V1 and some parts of V2 and other extrastriate visual areas: this peel represents the entire posterior occipital pole of the brain.

this problem, we note that we have complete information regarding connectivity of voxels in adjacent digital sections of a brain peel. Each slice through a peel yields contours which can then be linked to contours in adjacent slices using the voxel connectivity data. We can detect critical levels (sections containing critical points) by their attendant changes of topology. By following these contours between the topological transitions which mark the critical points, we are able to “parse” the brain surface into a series of generalized cylinders, which are composed of simple loops, and critical points, where generalized cylinders are created, merge or vanish.

There are theorems in differential topology (Wallace 1968) which state that the critical points of a differentiable surface ¹. are isolated and of only a few basic types. The

1. Our surfaces are substantially over-sampled, since we use 40 voxels to represent brain surfaces whose radius of Gaussian curvature is typically in the range of 1mm - 10mm. In our experience, the assumption that our voxel surfaces represent discrete samples of a differentiable surface is reasonable.

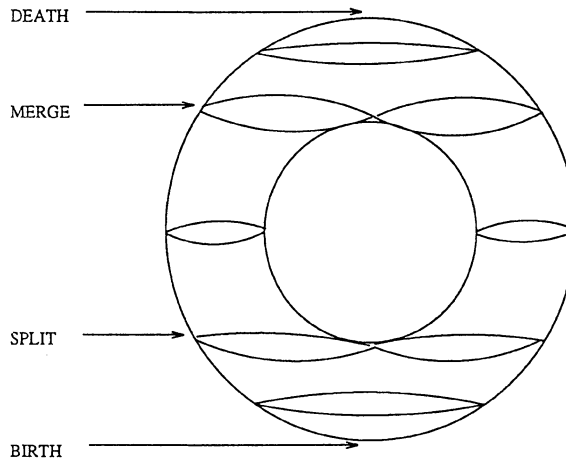


Figure 4 A torus, and the location of its four critical points: a maximum, two saddle points, and a minimum.

small number of possible types of critical points limits the kinds of topological changes which can occur between adjacent sections. Our strategy is to explicitly deal with each of the kinds of critical points that are possible (they are illustrated by the torus example), and thus be able to provide an argument for the correctness of the triangulation.

These ideas were recently implemented (Schwartz *et al.* 1988) by parsing a sequence of sections with connectivity data (i.e. a brain peel) into a series of generalized cylinders joined by critical points. This program acts as a translator which then constructs a program for a simpler triangulation algorithm which is capable of joining single loops. There are many such algorithms. The results of application to a peel of monkey brain are shown in Figure 5. This triangulation was accomplished without any human interaction.

Removal of human interaction is an important advance. The brain peeler algorithms provide about 20 brain peels per cortical area. Interactive triangulation of all this data is expensive and error prone. It is very important that our polyhedral models not be subject to arbitrary heuristics in the triangulation stage: we need correct models, in order to perform an accurate unfolding of the brain.

Flattening the brain

We have so far dealt with the cortex in its natural geometric setting: it is a curved two-dimensional surface. If we do not unfold the brain, we cannot observe much of its surface, which is hidden within the sulci of the cortex. It is also more difficult to construct models of neural architecture on curved surfaces than on planar surfaces. We have shown in earlier work that the curvature of V1 is not so large that we cannot represent it

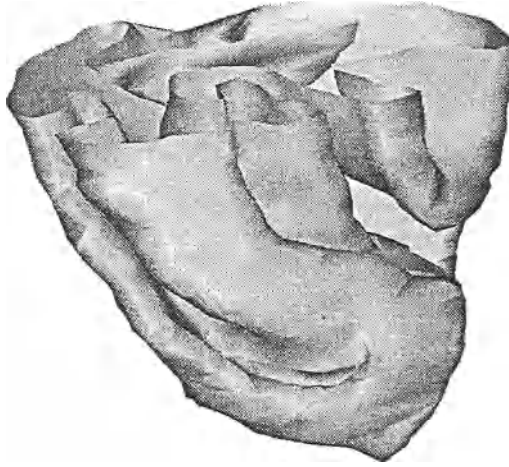


Figure 5 A polyhedral model of the brain which was reconstructed from serial sections. The sections were passed through the brain peeler to create a set of contours. These contours were parsed into generalized cylinders and critical points, and then connected by writing a program for a simple triangulator, using no human interaction. There are 1776 triangles in this model.

with a planar model. The values of both mean and Gaussian curvature of V1 (Schwartz and Merker 1986a), indicate that planar models of monkey cortex are acceptable. We therefore address the problem of flattening the brain.

There are two aspects to our method of flattening the brain. First, we specify a variational problem. Given a polyhedral model of the cortex, we generate a random set of points which are constrained to lie in the plane, one point for each node of the 3D model. We then set up two matrices of interpoint distances: one in the plane and one in the polyhedron. We define a least-squares goodness of fit between the two distance matrices. We shift the points in the plane, via a standard gradient descent scheme, to maximize the goodness of fit between the matrices. A good degree of fit indicates a good approximation to an isometry: a map from the 3D polyhedron to a 2D point set which preserves interpoint distances.

This algorithm, which is variously known as non-linear mapping or multi-dimensional scaling (Sammon 1969), works quite well with the type of data we are discussing here. We have found (Schwartz *et al.* 1989) that a mean local error of 5% is achievable by this process, and that subsequent randomization of the 2D point set leads to the same final configuration, indicating that the solution does not become trapped in local minima. The main difficulty in this scheme is to come up with the interpoint distances inside the 3D polyhedron. One would think that the problem of finding discrete

geodesic distances on the surface of (a nonconvex) polyhedron would have been solved long ago. But only in the past several years has a polynomial time algorithm been described for this problem (Mitchell *et al.* 1987). As far as we know, this very difficult algorithm has not yet been implemented. We have developed a simpler exponential time algorithm, implemented it, and found that for polyhedra with several thousand nodes, this algorithm has adequate performance on current work-station architectures. Our geodesic distance algorithm is described in (Wolfson and Schwartz 1989). Figure 6 shows an example of a flattened wire-frame of V1.

The flattening algorithm described above provides a planar model of a cortical area, such as that of V1, which has metric properties close to that of the original 3D surface. Although this procedure is intricate, it provides high precision data to be used for modeling of columnar and topographic architectures in cortex. Having produced both the flattened and 3D voxel and polyhedral models, the only remaining step is to texture map the gray-scale values from the 3D surface into the flattened model.

Texture Mapping

There is a one-to-one correspondence between the triangles making up the 3D brain surface, and the flattened 2D model; the shape of these triangles differs very slightly, as a result of the small (typically <5%) flattening error. In order to map the gray-scale values (i.e. the image of the ODC system in the 3D surface, as in Figure 3) into the planar model, we only have to perform an image warp of triangular element (from 3D) to triangular element (in 2D). This texture mapping takes the form of a bilinear warp, and results in a planar image of the surface of the cortical ODC pattern (Figure 1). We have thus obtained a flattened image of the ODC pattern in V1. Similar methods have been applied in our lab to the reconstruction of other features of cortical architecture (e.g. topographic mapping) and we believe that these methods are sufficiently general to be applied to a wide range of problems in computational neuroanatomy.

2. Level 2: Parametric Modeling of the ODC Architecture

The previous section described a system we have recently constructed whose purpose is to provide high resolution computer models of cortical architecture. A number of difficult algorithmic and implementation problems were encountered. Much support software for low level aspects of this work had to be constructed; the entire system represents many tens of thousands of lines of C code. However, the output of this system is merely a set of images. It is interesting to be able to inspect such images, but they have no meaning in and of themselves. We have stressed in this and other work that computer display is merely the first in several coordinated steps. In order to begin to understand a pattern, such as the ODC system, it is necessary to model it. This modeling effort is strongly constrained by the data provided in the first stage. As workers in computer graphics well know, images are unforgiving of hand-waving!

The ODC pattern bears a strong resemblance to other natural patterns, including the stripes of a zebra, the pattern of sand dunes in a desert, the domains in a magnetic bubble system, and wind-driven ocean waves. Some of these phenomena have proven amenable

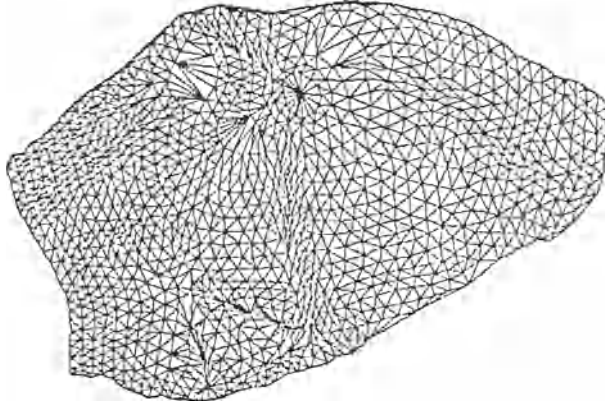


Figure 6 The flattened wire frame used to produce Figure 1.

to modeling via two-dimensional bandpass-filtered noise (e.g. (Mastin *et al.* 1987)). We apply a similar approach below.

This image-oriented approach contrasts with that of other workers, who have sought to construct explicit neural models of pattern generation (e.g. (Miller *et al.* 1989; Swindale 1980; Swindale 1982)). We avoid construction of a mechanistic model of the system, in favor of developing a simple image-oriented model which reproduces the observed patterns. Our approach has the following strong points:

- We can obtain maximum insight into the nature of the pattern itself by abstracting its properties in a simple model. We focus on the pattern itself, rather than on poorly understood processes in the developing nervous system.
- By avoiding the computationally expensive details of a mechanistic model, we can develop a rapid simulation tool which allows us to extensively explore the model.
- By abstracting the nature of the pattern in the form of a simple generative model, we can publish a parameterized procedure for generating these patterns, which spares workers wishing to reproduce the pattern the need to implement a complex simulation.

We have found that an extremely simple model is capable of reproducing the properties of the monkey ODC (and the cat ODC, as well as other columnar systems); this model is based on the properties of band-pass filtered white noise.

Bandpass-Filtered Noise

In two dimensions, bandpass filters may be classified according to whether or not they have a preferred orientation. When a filter has no preferred orientation, we say it is isotropic. In this case, the filter has an annular shape in the frequency domain (Figure 7a); in the space domain we find a kernel which has a familiar center-surround structure (Figure 7b). The filter is characterized by two parameters, the center frequency and the width. Filtering and thresholding a noise image with an isotropic filter yields a faintly columnar structure (Figure 7c, d) which lacks the striking medium-range directional correlation of the observed ocular dominance column structure.

When we turn to oriented (anisotropic) filters, we find that the frequency domain representation has humps centered on symmetric points in the frequency domain (Figure 8a), corresponding to planar cosine waveforms of particular frequency and orientation. The space-domain kernel of the filter consists of alternating positive and negative regions in the direction perpendicular to the preferred orientation, with the regions elongated parallel to the preferred orientation (Figure 8b). Filtered noise assumes a wave-like pattern (Figure 8c); after thresholding, we obtain a figure (Figure 8d) with a strong resemblance to the ocular dominance pattern at small scales.

The anisotropic filter is slightly more complicated than the isotropic filter; we require three parameters for a complete description. Like the isotropic filter, the center frequency and width must be specified. In addition, there is an orientation. The effects of varying these parameters are easy to describe. Increasing the center frequency decreases the column width. Changing the orientation of the filter alters the direction of the columns. Increasing the width of the filter increases the “noisiness” of the columns, resulting in more variation of column width and orientation, and increased branching and column termination.

Analysis of the Pattern of Ocular Dominance

The small-scale regularity of the pattern of ocular dominance is obvious; casual observation reveals that in any small region of the image, the columns are roughly parallel with relatively constant width. This leads us to expect that small subimages will have strong spectral components with direction and frequency consistent with the local direction and width of the columns. We observed exactly this phenomenon. We measured the orientation and frequency of the peaks in the normalized spectra of overlapping subimages, obtaining the variation of these parameters over the flattened brain region. Although there was considerable local variation of these parameters, we attribute most of the variation to poor resolution of the data. After smoothing the measured parameters, we concluded that the variation of column width (center frequency) was largely constant over the brain region, while the variation of orientation was smooth and consistent over large subregions, and probably correlated with gross brain anatomy.

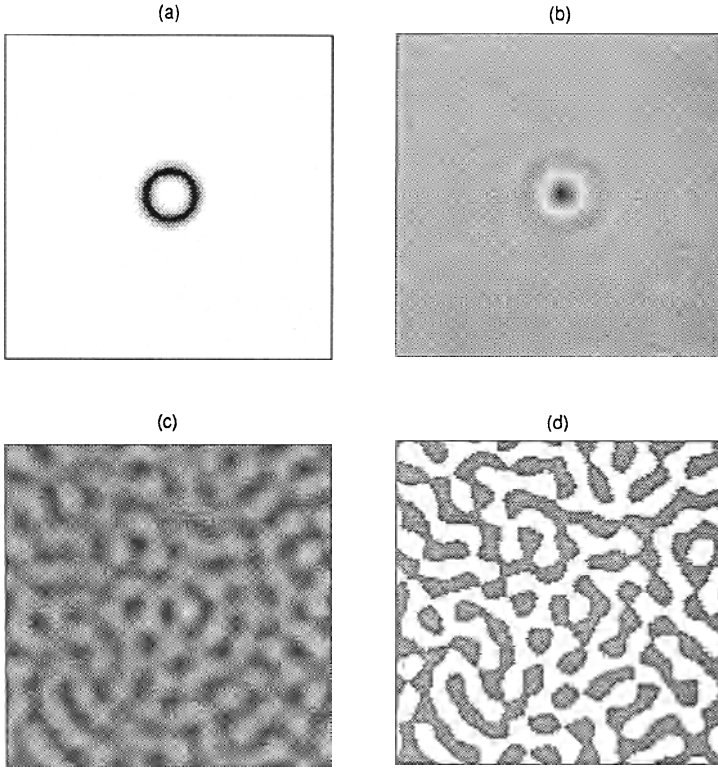


Figure 7 Two dimensional demonstration of the qualitative characteristics of unoriented (isotropic), bandpass-filtered noise, shown as intensity plots. (a) A typical isotropic bandpass filter in the frequency domain ($f_c = 0.15$, $\delta = 0.10$). (b) The (space domain) kernel of the filter shown in (a). (c) The convolution of the kernel in (b) with zero mean, unit standard deviation Gaussian noise. (d) The “columns” which result when the signal in (c) is thresholded.

Synthesis of the Pattern of Ocular Dominance

Although the spectra of the subregions which we analyzed were complicated, we found that we could easily obtain subjectively satisfying synthetic patterns. We constructed for each subregion an anisotropic bandpass filter, with the filter center frequency and orientation set to reflect the characteristics of the subregion’s spectral peaks, which we obtained from the measurements described above. We somewhat arbitrarily fixed the width parameter to $2/3$ of the center frequency, since this yielded satisfactory results. We applied this filter to Gaussian noise and thresholded the result; a typical example is shown in Figure 9.

Although this method yielded a good model of the column pattern on a small scale, we needed further refinements to apply the technique to the entire pattern. We

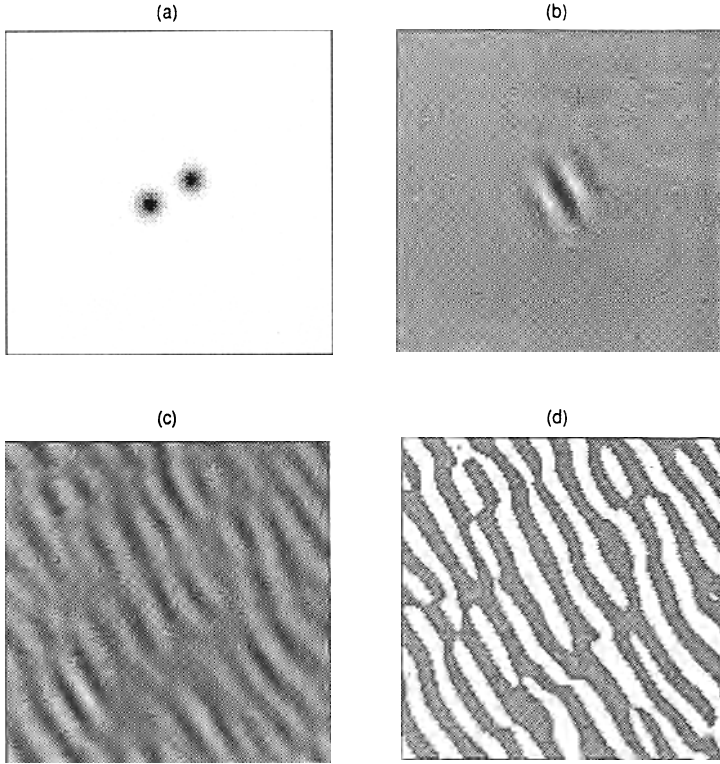


Figure 8 Two dimensional demonstration of the qualitative characteristics of oriented (anisotropic) bandpass-filtered noise, shown as intensity plots. (a) The frequency domain representation of an anisotropic bandpass filter with center frequency = 0.05, bandwidth = 0.08. (b) The (space domain) kernel of the filter shown in (a). (c) The convolution of the kernel in (b) with zero mean, unit standard deviation Gaussian noise. (d) The “columns” which result when the signal in (c) is thresholded. A typical isotropic bandpass filter in the frequency domain ($f_c = 0.15$, $\delta = 0.10$). (b) The (space domain) kernel of the filter shown in (a). (c) The convolution of the kernel in (b) with zero mean, unit standard deviation Gaussian noise. (d) The “columns” which result when the signal in (c) is thresholded.

constructed a “seed” noise image in register with the flattened brain image. We moved a window over the two images, applying to the underlying noise seed the parametric filter that was obtained from the smoothed parameters measured on the corresponding subimage in the actual ocular dominance pattern. As described above, this required the extraction of two parameters (orientation and center frequency) from the subimage of the actual ocular dominance pattern and a smoothing operation on the parameters thus determined.

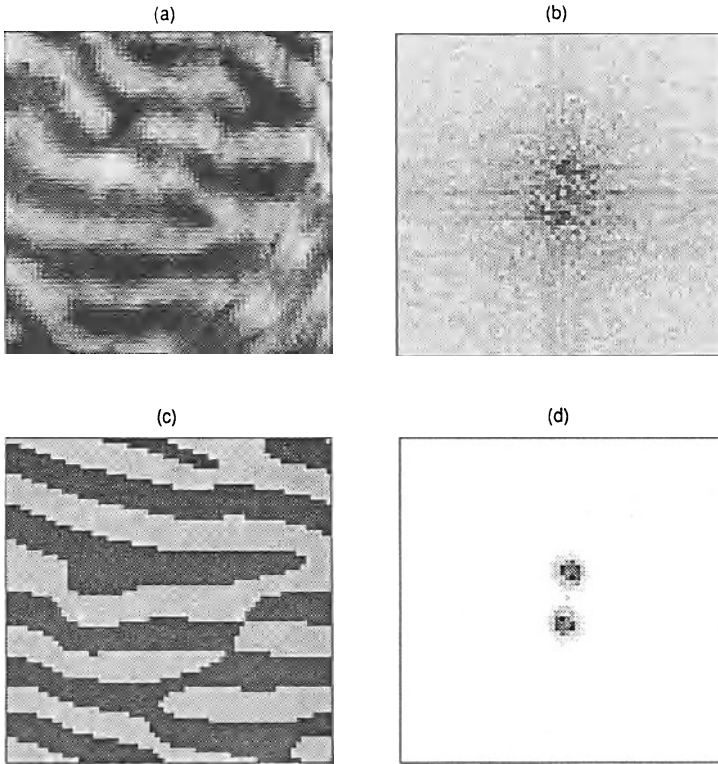


Figure 9 Examples of subimage analysis from the flattened brain. (a) A typical “good” subimage. (b) The spectrum of (a). From this spectrum, we derived filter parameters $f_c = 0.157$, $\theta = 84$ deg. We also use $\delta = 0.67f_c$ and $\epsilon = 1$. (c) An image synthesized by application of the filter parameters derived from (b) to Gaussian noise followed by thresholding. (d) The spectrum of the synthetic image before thresholding (i.e. the convolution of the derived filter with a Gaussian noise).

We then extracted a subimage from the synthetic pattern, which was used to “tile” a synthetic pattern of ocular dominance corresponding to the entire measured pattern. Lastly, the synthetic pattern was smoothed with a median operation. The result is shown in Figure 10.



Figure 10 Comparison of actual and synthetic column data. (a) Actual column data from the flattened brain. (b) Synthetic columns generated with parametric filtering and blending.

We can summarize the synthesis of the ocular dominance pattern:

Algorithm ODC_synthesis

```

For each position of the movable window
    extract the local column pattern from the flattened brain data
    determine the spectral parameters (center frequency and orientation)
Smooth the measured parameters
Generate a noise seed in register with the brain data
For each position of the movable window
    extract the registered noise subimage from the noise seed
    construct a filter according to the smoothed spectral parameters
    filter and threshold the noise subimage
    extract the central region of the subimage
Combine all the extracted regions in their proper positions
(* these regions are non-overlapping and exhaustively cover the original *)
Smooth the synthetic columns
  
```

Synthetic Columns as a Model for Ocular Dominance

The synthetic columns which we have constructed seem to successfully capture the qualitative characteristics of the actual pattern of ocular dominance. The model embodies the strong points which we have mentioned earlier. This system is extremely fast (we can generate a high-resolution model of the full primate ODC on a workstation in several minutes). It is a simple parameterization which other workers can easily use.

The ability to economically simulate columnar systems with high accuracy is of great importance to further modeling of the properties of visual cortex.

We believe that the properties of filtered, thresholded noise underlie the mechanisms of all of the related column systems (zebra stripes, magnetic bubbles, sand dunes and ocean waves). The kernels associated with the anisotropic bandpass filters are probably an essential component of any realistic model of primate ocular dominance, whether they are present as oriented receptive fields or correlated input. An underlying noise source is also necessary to allow bifurcation of symmetric initial conditions into two sets of columns (symmetry breaking). The threshold operation captures the “positive feedback with bounds” that is implicit in models which modify connectivity according to correlated activity yet do not permit unbounded growth of connection strengths. Thus our model highlights the essential properties that are required to generate the ocular dominance structure, without the need to specify low level developmental processes.

3. Level 3: Computational Significance of the ODC Architecture

In the preceding sections, we presented a detailed discussion of the measurement and parametric modeling of the ODC system. We have not yet considered the function of the system; what possible computational significance might there be to formatting binocular data in the form of the ODC system?

We will review recent work (Yeshurun and Schwartz 1989) which describes a simple nonlinear filter (the *cepstrum*) which can use a columnar data format to extract binocular disparity between paired images. In this work, we show that the column/cepstrum model has properties consistent with the limitations of stereo vision in humans.

Stereo Segmentation and the ODC pattern

The lateral offset of the two eyes introduces differences between the two images which are projected to the cortex. The positional and orientational shift of an image feature from one image to the other is the *binocular disparity* of the feature. Many algorithms to compute binocular disparity have been proposed over the past decade or so. Some algorithms attempt to locate salient features in each image, and then match them between images. These algorithms suffer from the ambiguity of multiple instances of the same feature in an image. Other algorithms systematically shift one image to try bring it into alignment with the other image (correlation). These algorithms are computationally expensive; when there are N pixels in an “eye” or image to be processed, direct correlation will involve computation of order of N^2 .

In the ODC system, we observe that data from the two eyes is formatted in paired stripes. This suggests an algorithm utilizing interleaved data from the two eyes. The columnar layout can be characterized as a kind of “visual echo”: a small patch of the left eye view of the scene is echoed in the right eye view with a shift of one column width plus the binocular disparity we are trying to measure. A well known and highly effective algorithm for echo detection in auditory processing is the cepstral filter (Bogert *et al.* 1963). The cepstrum is the power spectrum of the log of the power spectrum of an signal. Because of the two power spectra, the units of the cepstrum revert to the domain

of the original signal. For an audio signal, this means the cepstrum has units of time. It can be shown that the cepstrum of an auditory signal with an echo will have a prominent peak at a position corresponding to the time delay of the echo.

To explore this idea, we implemented a two-dimensional version of the cepstrum and applied it to pairs of images formatted as small paired patches. The relative scale of the images and patches were chosen to be consistent with normal viewing conditions, and the anatomical scale of the ODC system. In Figure 11a, a gray-scale image which subtended about 8 degrees of arc is shown. Each subimage shown in Figure 11b is about one column width (5 minutes of arc). The data is shown as two copies of the same section of the large image, taken from the position of the arrow. The cepstrum (Figure 11c) applied to this pair gives a strong peak at a position corresponding to a shift of one column-width. In Figure 12, we demonstrate a stereo segmentation on a random dot stereogram of large size (3500x3500), which contains a pac-man figure as correlated binocular cues. We ran the cepstrum on small windows corresponding to square patches of the size of ocular dominance columns. This approach produced a rapid segmentation of this extremely large binocular stereo problem (Yeshurun and Schwartz 1989).

From a computational point of view, the column/cepstrum algorithm has very good complexity: it is $N \log N$ in the number of pixels. From a biological point of view, the algorithm can be implemented by a system of neurons which is capable of extracting power spectral estimates of the "cortical image". But this is one of the basic features of cortical neurons: they are medium-band spatial filters, with bandwidths typically of 1.5 octaves. In other work, we have shown that a two dimensional cepstrum can be effective for stereo segmentation when implemented either with a standard FFT algorithm (computational application) (Yeshurun and Schwartz 1989) or with medium bandwidth spatial filters (neural application) (Yeshurun and Schwartz 1988).

There are a number of interesting consequences of the column/cepstrum algorithm, which are related to its columnar data format. Because of the windowed nature of this algorithm, it cannot analyze very rapid changes in disparity (i.e. varying over scales smaller than the column width). There will be natural scale factors in behavioral aspects of stereo vision which are set by the size of ocular dominance columns. We have discussed these characteristic properties elsewhere (Yeshurun and Schwartz 1988) Yeshurun cepstrum ; where we show that there is a correspondence between these predicted algorithmic properties and the psychophysical details of human binocular stereo vision.

In summary, we have shown that the ODC pattern of interlaced inputs from the left and right eyes has an intriguing relationship to the functional aspects of stereo vision, to which this system is almost certainly deeply related. From a computational point of view, cepstral stereo has very attractive performance aspects. Recently, Ballard and Brown have implemented the cepstral algorithm in hardware, and have used it to control the binocular vergence of a machine vision binocular camera system.

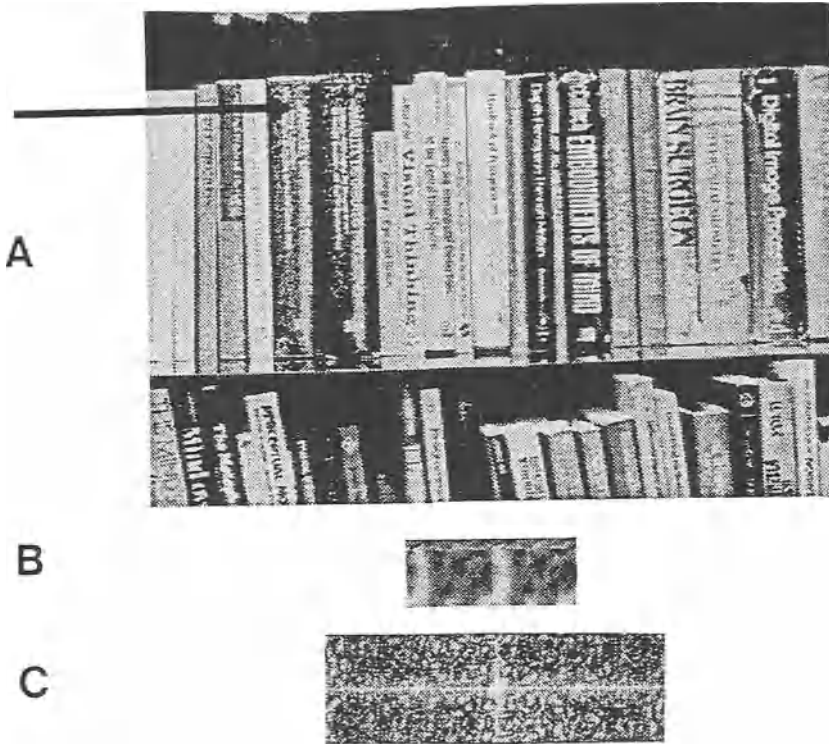


Figure 11 (a) A natural scene, which was photographed with a stereo camera technique. (b) The arrow points to a tiny portion of the scene, comprising about 5 minutes of arc, which is reproduced in a magnified form in Figure 11b. (c) The cepstrum of B above, which represents a small window of a stereo frame, simulating the means by which this data would be presented in the brain via the ocular dominance column system, i.e. as a pair of small patches of (left/right) image, with a small disparity component of shift added to the normal columnar offset. The bright peaks in the figure represent the magnitude of the binocular disparity (modulo the column size).

Conclusion and Prospects

This paper has developed several themes related to the visualization and conceptualization of patterns of functional architecture in the mammalian brain. In order to visualize a brain pattern, one must first deal with the extremely challenging problems associated with reconstructing it from raw data. We described some half-dozen algorithms which we developed for this purpose. We illustrated their application to a single architectural feature of primate visual cortex, the ocular dominance column system of V1. We showed that bandpass-filtered thresholded white noise provides an economical graphical model of this pattern. We outlined an algorithmic justification for the presence of

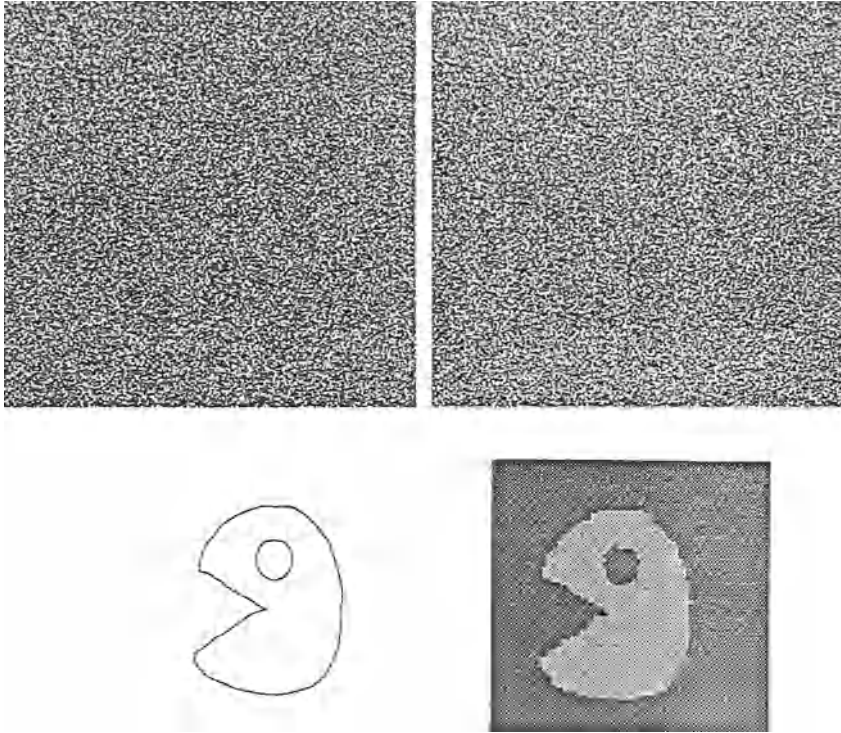


Figure 12 A segmentation of a simulated 8 by 8 degrees RDS (3500 by 3500 pixels) by the algorithm. The figure can be segmented only by binocular disparity cues. The original image consists of 3500 by 3500 pixels, that represent a 8 degree image. Taking windows of 32x32 pixels each, the method produces a segmented image of 100 by 100 pixels, simulating 5 minute windows over the entire image.

columns in a visual system which computed binocular disparity; in conjunction with a simple nonlinear filter (the cepstrum), we obtained a fast and robust algorithm for stereo segmentation.

Application to other neural systems

In addition to the ODC system, there are many other examples of columnar architecture in the brain. For example, neurons which respond to oriented edges of similar orientation are grouped together into “orientation columns” in V1 and neurons which respond to a similar direction of movement are grouped together into “direction columns” in a cortical area specialized for motion (area MT). Columnar grouping has been observed in frontal cortex, and secondary visual cortex (V2) has at least three

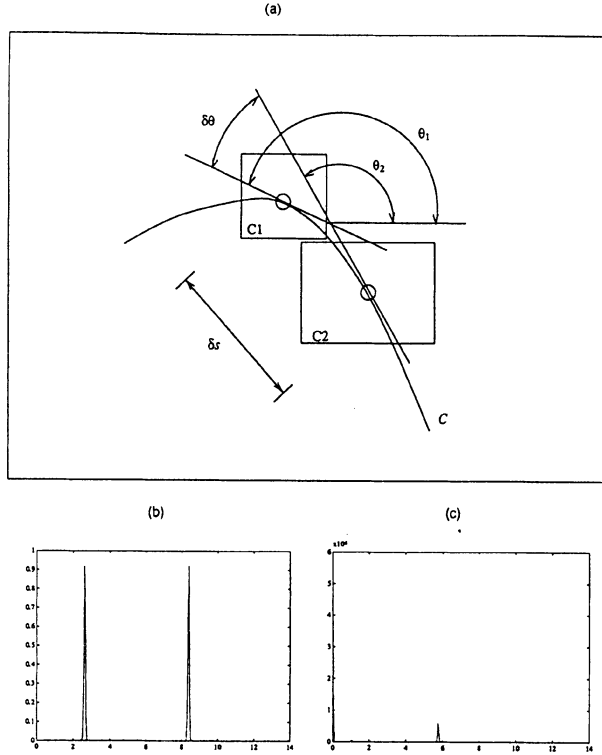


Figure 13 Application of the cepstrum to curvature computation. (a) Geometry of the model. A contour C passes through the receptive fields of two “hypercolumns”, shown here as boxes $C1$ and $C2$. The contour has approximate orientation $\theta_1 = 2.618$ radians in $C1$, and $\theta_2 = 2.094$ radians in $C2$. The approximate curvature of the contour in the region of $C1$ and $C2$ is given by $\delta\theta/\delta s$, with $\delta\theta = \theta_2 - \theta_1 = -0.524$ radian and δs the arc length shown. (b) Representation of θ_1 and θ_2 as distributions of activity along adjacent hypercolumns. (c) Positive part of cepstrum of the signal in (b). Note the presence of a peak at 5.7596 radians, corresponding to $\delta\theta + 2\pi$.

interlaced column systems. This list will undoubtedly expand with continued research. We feel that the presence of columns, that is, of groups of neurons in periodic patterns on a scale that is large compared to the size of single neurons, is one of the basic architectural schemes of neocortex.

Many workers have speculated about the existence of these column systems: why are they so common? The results summarized in this paper suggest two answers: columns are easily constructed and they are computationally useful. We have shown that the generation of the columnar pattern can be accomplished by an almost trivial mechanism: bandpass filtering followed by thresholding. The space-domain dual of bandpass filtering is correlation. Our studies thus suggest that any mechanism which introduces local correlation of form dual to the bandpass filters which we have described will result

in column systems of the kind actually observed in the brain. We should not be terribly surprised to see columns: they are one of the simplest patterns to build!

The ease of construction couples nicely with the functional utility of column systems. When one wishes to extract the difference of two similar signals, the column/cepstrum architecture is ideal. Using the notion of generalized difference mapping, we can suggest another application to processing of visual information. Sensitivity to visual “edges”, i.e. abrupt changes of contrast, is also formatted across the surface of cortex in a periodic fashion, in the form of “orientation columns.” V1 comprises a periodic orientation map, as well as a periodic stereo map. Changes in orientation are proportional to curvature, and curvature is one of the principal shape descriptors which the visual system must analyze. In Figure 13, we briefly illustrate how the cepstral operator, when applied to orientation columns, yields a strong peak in a position approximating to the local curvature of a contour. The important point is that the same column/cepstrum architecture used for computation of binocular disparity can be applied to the computation of contour curvature, illustrating the power of a generalized difference map. This economy of architecture may be the prototype of a generic computational module in neo-cortex.

The orientation column pattern may be generated by thresholding isotropic bandpass-filtered noise. The issues involved in visualizing and measuring the orientation column system are similar to those outlined above for the ODC system. All of the methods and demonstrations described for the ODC system could be applied to the orientation column system, given appropriate raw data. The same may well hold true for the other column systems of the cortex.

The principle theme of this work is that cortex is a highly patterned “machine”, whose purpose is to recognize patterns. Our goal has been to uncover general cortical mechanisms by application of the methods of computer graphics and numerical modeling to visualize those patterns which are the basis of our ability to visualize patterns.

References

- Anjyo, K., Ochi, T., Usami, Y., and Kawashima, Y. (1987), A practical method of constructing surfaces in three-dimensional digitized space, *The Visual Computer* 3 (1): 4-12.
- Artzy, E., Frieder, G., and Herman, G.T. (1981), The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm, *Computer Graphics and Image Processing* 15: 1-24.
- Bogert, B.P., Healy, W.J.R., and Tukey, J.W. (1963), The frequency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking, *Proc. Symp. Time Series Analysis*, John Wiley and Sons, New York, pp. 209-243.
- Frederick, C., and Schwartz, E.L. (1990), The brain peeler: viewing the inside of a three dimensional shell, *Visual Computer* 6(1): 37-49.
- Fuchs, H., Kedem, Z.M., and Uelson, S.P. (1977), Optimal surface reconstruction from planar contours, *Communications of the ACM* 20: 693-702.

- LeVay, S., Hubel, D.H., and Wiesel, T.N. (1975), The pattern of ocular dominance columns in macaque visual cortex revealed by a reduced silver stain, *J. Comp. Neurol.* **159**: 559-576.
- Mastin, G.A., Watterberg, P.A., and Mareda, J.F. (1987), Fourier synthesis of ocean waves, *IEEE Computer Graphics and Applications* **7**: 16-23 (March).
- Miller, K.D., Keller, J.B., and Stryker, M.P. (1989), Ocular dominance column development: analysis and simulation, *Science* **245**: 605.
- Mitchell, J.S.B., Mount, D.M., and Papadimitriou, C.H. (1987), The discrete geodesic problem, *SIAM J. Comput.* **16**: 647-668.
- Sammon, J.W. (1969), A nonlinear mapping for data-structure analysis, *IEEE Trans. on Computers* **C-18**: 401-409.
- Schwartz, E.L., and Merker, B. (1986), Computer-aided neuroanatomy: differential geometry of cortical surfaces and an optimal flattening algorithm, *IEEE Computer Graphics and Applications* **6**(2): 36-44 (March).
- Schwartz, E.L., and Merker, B. (1986), Three-dimensional computer reconstruction of the ocular dominance column pattern of macaque striate cortex: a digital tangential microtome, *Invest. Ophthal. and Vis. Sci.* **27**: 223.
- Schwartz, E.L., Merker, B., Wolfson, E., and Shaw, A. (1988), Computational neuroscience: Applications of computer graphics and image processing to two and three dimensional modeling of the functional architecture of visual cortex, *IEEE Computer Graphics and Applications* **8**(4): 13-28 (July).
- Schwartz, E.L., Shaw, A., and Wolfson, E. (1989), A numerical solution to the generalized mapmaker's problem, *IEEE Trans. Pattern Analysis and Machine Intelligence* **11**: 1005-1008.
- Schwartz, E.L., (1990), Introduction, in: *Computational Neuroscience*, E.L. Schwartz ed. Cambridge, MA: MIT Press.
- Swindale, N.V. (1980), A model for the formation of ocular dominance column stripes, *Proc. Roy. Soc. Lond. B* **208**: 243-264.
- Swindale, N.V. (1982), A model for the formation of orientation columns, *Proc. Roy. Soc. B.* **215**: 211-230.
- Wallace, A.H. (1968), *Differential Topology*, New York: W. A. Benjamin.
- Wolfson, E., and Schwartz, E.L. (1989), Computing minimal distances on arbitrary polyhedral surfaces, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **11**: 1001-1005.
- Yeshurun, Y., and Schwartz, E.L., (1988), Ocular dominance columns in macaque V1: a two dimensional stereomap, in: *Computational Neuroscience*, E. Schwartz ed. Cambridge MA: MIT Press.
- Yeshurun, Y., and Schwartz, E.L. (1989), Cepstral filtering on a columnar image architecture: a fast algorithm for binocular stereo segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence* **11**(7): 759-767.

DYNAMIC VISION

Harry Wechsler* and Lee Zimmerman**

*Computer Science
George Mason University
Fairfax, VA 22030
USA

**Electrical Engineering
University of Minnesota
Minneapolis, MN 55455
USA

1) Introduction

The fundamental problem faced by all vision systems is the ambiguity created by the projection process. An object's projected shape in an image changes dramatically for small changes in the observer's viewpoint. This is the basic difficulty in creating a machine vision system that can respond robustly in an unconstrained 3D environment. Our approach to this problem enables the vision system to actively engage its interpretation of the surroundings using a distributed memory system modeled as visual potentials and made up of characteristic view(point)s (Koenderink and van Doorn, 1979).

The suggested vision system consists of a feedback loop. Recognition affects depth interpretation and depth interpretation affects recognition. Input to our system is a 2D image and a sparse depth map (a viewer-centered map of the distances to visible surface points). A sparse depth map can be obtained from locality systems, which are modular units that use cues as stereo, shading, or motion to make estimates of depth (the distance from the observer to each point in the image). The output from the system is a dense map and the corresponding classification of the visible surfaces. Our system is a dynamic 3D image interpretation system in which location and recognition work hand in hand to produce solid and stable results.

The vision system consists of four main components, which are graphically displayed in Fig. 1. It begins with the formation of the depthmap. The depth map formation system blends information from visual distance cues such as motion or stereo, and recognition coefficients from the memory, appropriately 'filtered' through the active perception system, to create a smooth dense map. The depth plane is the result of the relaxation of a thin plate (Terzopoulos et al, 1987) across high confidence distances. The intensity image and the depth map are used by the reprojection system to produce a distorted image - a flattened, frontal version of all visible surfaces. This flattened image is then analyzed by the memory system. The distributed associative memory (DAM) system used is similar to the rotation and scale invariant memory system described by Wechsler and Zimmerman (1988). High confidence from the recognition system for an area of the image will solidify the depth interpretation in that area. Characteristic views (aspects) are stacked together within the (DAM) memory and the dynamics of the visual system engage in active exploration to seek that view needed to enhance the interpretation process. The volume of space being examined by the reprojection system, the scale

of the features, and the stability of the depth map are under the guidance of the active perception (Bajcsy, 1988) system. The way in which the active vision system will alter the interpretation process is by choosing new viewpoints to be examined.

2) DEPTH MAP FORMATION

The depth map formation system uses a relaxation process to blend location and recognition information. Often this information is sparse or imprecise across the field of view, so it is necessary to employ relaxation in order to interpolate and smooth. The depth map formation system consists of three planes -- confidence, distance, and thin plate or membrane, which are in registration with the input image. The thin membrane is the dense depth map composed from the anchor points, held in the distance plane, and initialized by the locality systems.

The confidence plane receives information from the memory (filtered through the active perception system) and the locality systems. The locality systems produce sparse distance information, irregularly spaced across the field of view. Each point in the confidence plane indicates the confidence in the perceived distance value. When confidence is high the thin membrane is frozen and not allowed to change. When confidence is low the thin membrane can change a great deal. The distance plane contains the anchor points used by the relaxation process. High confidence from the recognition system for an area of the image will solidify the depth interpretation in that area. The depth map is formed from sparse depth information in the following way. First, the data is interpolated using a thin plate. The energy function used to enforce the constraints among the grid points is similar to that used to implement *snakes* (Kass et al, 1987). The energy function is composed of two terms. One enforces the constraint that the derived depth map be close to the actual range data; the other tries to find the flattest membrane that will fit the data. The stiffness of the plate is slowly reduced resulting in a membrane interpolation that captures the surfaces indicated by the data. This is done using successive relaxation steps.

3) REPROJECTION

When an object rotates about an axis perpendicular to the image plane, its projected shape will rotate but will not change shape. On the other hand, if the same object rotates about an axis parallel to the image plane (rotation in depth), its projected shape will change dramatically due to the appearance of new surfaces, foreshortening, and perspective distortions. Foreshortening occurs when the surface being viewed slants away from the viewer. Perspective projection occurs in all imaging devices and results in parallel lines on the surface becoming converging lines in the image plane. Our computer vision system tries to minimize this problem by 'reprojecting' the surfaces being viewed to a flat, frontal position. The reprojection system must satisfy several criteria. First, the reprojection function should compensate for the distortion of a single disconnected planar surface. This requires that surface markings, such as writing, be invariant to the slant of the surface. Second, the reprojection function should topologically map an object in an image to a characteristic shape. Third, the reprojection function should have

the ability to zoom in on an area of the image -- limiting the spatial extent of the processing. Following the above criteria, the whole system should be able not only to recognize an object from its characteristic surface shapes but should also be able to recognize surface markings.

The reprojection function assumes each pixel is an area of intensity at a distance given by the dense depth map (thin plate). The area of the pixel is expanded first horizontally and then vertically from the center axis outward. Each pixel is expanded according to the relative distance between itself and its neighbor along the direction of expansion. The reprojection will correctly compensate for the distortion of a single planar surface. It is non - linear and memoryless. Changes occurring to either the image or the depth map are immediately incorporated into the reprojected image.

Let the input information to the reprojection system be from a single plane slanted in space about the x - axis. The slanted plane will be vertically foreshortened in the image. If the plane were frontal, the vertical distance of the object's surface covered by the center two adjacent pixels is given by $h = d / f$, where h is the frontal vertical distance, f is the focal length, and d is the distance from the lens to the surface of the plane. The distance along the slanted plane covered by the center pixels is

$$v = d / [\text{fcos}(\theta) - \sin(\theta)]$$

where θ is the slant angle of the plane. The vertical distance v is larger than h for positive θ , smaller than h for negative θ , and equal to h for $\theta = 90^\circ$. To rotate the plane to a frontal position in the projection plane, the image needs to be expanded or contracted by the ratio v / h . For a surface of arbitrary slant and tilt, the image is expanded in the vertical direction and then expanded in the horizontal direction using gradient space (p , q) information.

The reprojection function is a homolographic projection of the visible surfaces in the image. Homolographic projections result in mappings where visible object surfaces maintain their relative 3D size. When this projection is performed on a world globe, the result would be equivalent to Lambert's Equal Area Map first developed in 1722. This mapping is still commonly used for polar regions.

Curved surfaces could be treated as multi - faceted planar surfaces. All visible surfaces can be approximated to within a finite error by planar surfaces if the size of the planar surface is small enough. Essentially, the reprojection function 'sees' each pixel as a planar frontal surface in 3D space, which has been projected to form the image being analyzed. The fundamental assumption for our computer vision system is that a 3D object is represented as a collection of 2D reprojections of the object from widely varying viewpoints.

4) MEMORY

The particular form of memory we deal here, related to classical conditioning or Hebbian learning, is of the DAM (Distributed Associative Memory) (Kohonen, 1987) type. DAMs are example of NN (Neural Networks) (Anderson and Rosenfeld, 1988) and/or PDP (Parallel and Distributed Processing) (McClelland and Rumelhart, 1986) models. Stimulus and response vectors are associated, and the result of association is spread over the entire

memory space. Parallel and distributed computation means that information about a small part of the association can be found and processed over a large area of the memory. New associations can be placed over the older ones and interact. The size of the DAM stays the same regardless of the number of associations that have been memorized.

The above discussion illuminates several properties of DAM which are different from the more traditional ones about memory. The associations are allowed to interact with each other and an implicit representation of structural and contextual information can develop. Consequently, a very rich level of relationships can be captured. There are few restrictions on what vectors can be associated. Consequently, extensive indexing and cross - referencing can develop. Furthermore, since the information is distributed across the memory, the overall function of the recognition system becomes robust to both memory faults and degraded stimulus vectors. The DAM operation includes memory construction and recall, and they are discussed next.

The construction stage assumes that there are n pairs of m - dimensional vectors that have to be associated. This can be written as

$$Ms_i = r_i \quad \text{for } i = 1, 2, \dots, n$$

where s_i and r_i denote the i th stimulus and response vectors, respectively. One seeks the memory matrix M such that when the k th stimulus vector s_k is projected onto the space defined by M the resulting projection will be the response vector r_k . Specifically, we have to solve

$$MS = R$$

where S and R are the corresponding stimulus and response matrices, respectively. A unique solution for the above equation does not necessarily exist for any arbitrary group of associations. The number of associations n is usually much less than m , the dimension of the vectors. Consequently, the system of equations is underconstrained. The constraint used to seek for a unique matrix M is that of minimizing the norm $\|MS - R\|^2$ which yields

$$M = RS^+$$

where S^+ is known as the Moore - Penrose generalized inverse of S (Kohonen, 1987).

The recall operation projects an unknown vector s onto the memory space M . The projection yields the response vector r ,

$$r = Ms = R(S^+s)$$

If the memorized stimulus vectors are independent, and the unknown stimulus vector s is one of the memorized vectors s_k , then the recalled vector will be the associated response vector r_k . If the memorized stimulus vectors are dependent, then the vector recalled by one of the memorized stimulus vectors will contain the associated response vector, and some *cross - talk* from the other stored response vectors. The recall can be viewed as the weighted sum of the response vectors. The recall, using a linear squares classifier, begins by assigning weights according to how well the unknown stimulus vector

matches with the memorized stimulus vector. The response vectors are multiplied by the corresponding weights (given as S^+s) and summed together to build the recall. The recall is usually dominated by the memorized response vector that is closest to the unknown vector. The DAM provides for interactions between the stored associations and thus allows for 'some' generalization, useful when facing stimuli the system has not been trained with beforehand.

The DAM is augmented using conformal mapping (Wechsler and Zimmerman, 1988) to provide for invariance to rotation and scale changes and the whole memory structure is shown in Fig. 2. Assume that pixels in the Cartesian plane are given as $(x,y) = [\text{Re}(z), \text{Im}(z)]$, where $z = x + jy$. One can then write $z = r[\exp(j\theta)]$, where $r = |z| = (x^2 + y^2)^{1/2}$, and $\theta = \arg(z) = \arctan(y/x)$. The conformal mapping is then the mapping of points z onto points w and it is defined as

$$w = \ln(z) = \ln[r(\exp(j\theta))] = \ln(r) + j\theta$$

Points in the target domain are given by $[\ln(r), \theta] = [\text{Re}(w), \text{Im}(w)]$. Logarithmically spaced concentric rings and radials of uniform angular spacing are mapped into uniformly spaced straight lines. (See Fig. 3.) If the scale and rotation factors are k and ϕ , respectively, then z_{old} and z_{new} are given by $r[\exp(j\theta)]$ and $(kr)\exp[j(\theta + \phi)]$, respectively. After conformal mapping, rotation and scaling about the origin become linear shifts in the $\theta(\text{mod } 2\pi)$ and $\ln(r)$ directions, respectively. The association that the DAM builds up is that between the magnitude and phase of the Fourier transform. The stimulus to DAM is the magnitude, and is invariant to both rotation and scale changes because those correspond to mere translation after conformal mapping. The response (and then recall) is the phase. The difference between the recorded and recalled phase indicates the amount of rotation and/or scale change.

Recognition takes place when processed information is matched with memory. When the correlation is high, both recognition and location of the object will occur. Our system assumes the matching, and thus the storage of models is done in terms of 2D views. The structure of the mapping implied by the proposed model is viewer - centered with the relative depth information placed implicitly within the stored model. As more dynamics are added to the system, these views will be dynamically 'tied' together to form the whole object, and will amount to viewer - oriented rather than viewer - centered representations.

The input to the distributed associative memory are the flattened characteristic views of the object. A characteristic view of a polygon corresponds to those point of view that have a specified number of planar surfaces present in the projected image. For example, a cube will have three characteristic views -- one side present, two sides present, and three sides present. The system is designed such that it will recognize the object and recognize the pattern printed on the object's surface.

The output from the memory is a classification vector, which measures how well the input view matches the stored views. It is used by the depth map formation system, appropriately 'filtered' by the active perception system, to adjust (raise or lower) the confidence plane.

5) ACTIVE PERCEPTION

The ultimate goal of our vision system is to interpret the scene regarding what is there and where things are. This requires a definite interaction between location and recognition. The best way to examine the system is to separate it into automatic and active parts. The automatic parts are functional and include the construction of the depth map, reprojection of the image, and recognition by the memory. The active components control the search process by specifying the volume of space to be analyzed by the reprojection system, choosing new fixation points, and determining the balance between reliance on cues from the locality systems and the need to smooth the thin membrane. The active perception system accomplishes this by changing the parameters of other systems using (memory) recognition information.

The volume of space under inspection will change within the parameters of attention point, depth point, and depth gradient. The attention point is the 3D point along the reprojected image's line of sight. The depth point is the point along the line of sight where the expansion or contraction of the reprojection function stops. The depth gradient is the maximum gradient marking where the expansion or contraction of the reprojection function stops as well. All three of them are actively determined. Notice that the volume of space being processed at any moment is determined by these parameters and by the environment under inspection, and it corresponds to the attention element of active perception.

The response from the memory is used to modify the shape of the depth map. The time course and extent of this modification is determined by the active perception system. For example, if the locality systems were sending information that the object under scrutiny is flat, but the memory system's best guess is that the object should be curved in a specific way, then after a certain amount of time the active perception would release the distance restrictions enforced by the locality systems. This is done by lowering the confidence in those distance estimates given by the locality systems. The reduction in confidence increases the importance of the recognition system allowing the depth map to become appropriately curved. The ability to override the different cues to depth is critically important both because the cues may be invalid for a variety of reasons and because the task may require it. One such task would be to recognize the person's face from a photograph. Stereo and motion cues could indicate the photograph is a flat surface with blotches of color on it, while the recognition system wants to recognize the 3D face that the photograph represents.

Finally, the active vision system will alter the interpretation process, choosing new viewpoints to be examined. If the recognition system cannot converge, then the system should physically change its position in space, and reinterpret the surroundings. Choosing new viewpoints corresponds to the exploratory aspect of active perception. In our 3D recognition system, the idea of active vision is not only a process of motor control, but is a process for engaging the environment in search for interpretations of the surrounding environment.

The experimental system has operated successfully on a database of seven viewpoints. The database used to create the DAM consists of one view of a cube, two views of a pyramid, one view of a plane, and three views of a tetrahedron. The recognition histograms, i.e., S^+ s, have indicated that the

system can correctly identify a novel input s , where the novelty is with respect to that input's attitude in 3D space.

6) CONCLUSIONS

The methodology suggested herein for 3D object recognition includes explicit visual dynamics to actively engage the environment and object-centered representations. The approach thus combines object and viewer-oriented representations through active perception. Our system, in effect balances object recognition with an understanding of the visible space.

The memory, of DAM type, can be organized in terms of visual potentials, where the characteristic views are 2D (intrinsic) views and the edges connecting them are related to exploratory events characteristic of active perception. Geometric distortions are accounted for by distributing responsibility between low- and high-level vision.

Note

A longer version of this paper, by the same authors and entitled 'Active perception and 3D object recognition', appears in *Active Perception and Robot Vision*, edited by A.K. Sood and H. Wechsler, NATO ASI Series F, Vol. 83, Springer-Verlag 1992.

References

1. Anderson, J.A. and E. Rosenfeld (Eds.): *Neurocomputing*, MIT Press 1988.
2. Bajcsy, R.: Active perception, *Proc. IEEE*, 76(8), 996 - 1005 (1988)
3. Kass, M., A. Witkin, and D. Terzopoulos: Snakes: Active contour models, *Int. Conf. on Computer Vision*, London, England, 259-268 (1987)
4. Koenderink, J. and A. van Doorn: The internal representation of solid shape with respect to vision, *BiolCyb*, 32, 211-216 (1979)
5. Kohonen, T.: *Self-Organization and Associative-Memories* (2nd. ed.), Springer-Verlag 1987.
6. McClelland, J., D. Rumelhart and the PDP Research Group (Eds.): *Parallel Distributed Processing*, MIT Press 1986.
7. Terzopoulos, D., A. Witkin, and M. Kass: Symmetry-seeking models for 3D object reconstruction, *Int. Conf. on Computer Vision*, London, England, 269-276 (1987)
8. Wechsler, H., and L. Zimmerman (1988), 2-D Invariant Object Recognition Using Distributed Associative Memory, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.10, No. 6, 811-821 (1988)

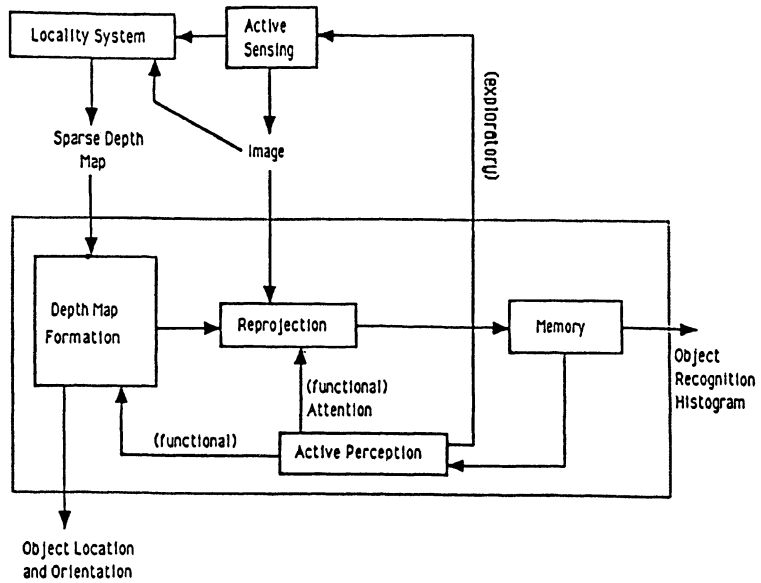


Figure 1. Block Diagram of the Wechsler and Zimmerman 30 Object Recognition System

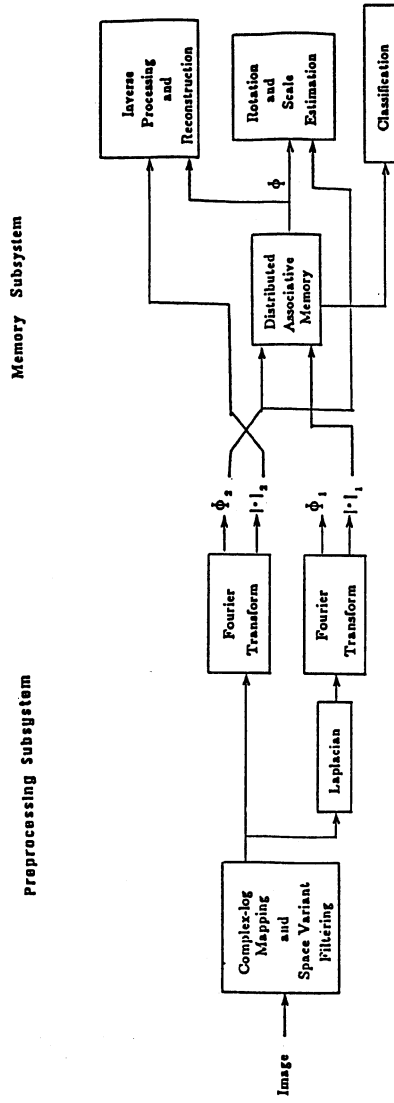


Figure 2. Memory System

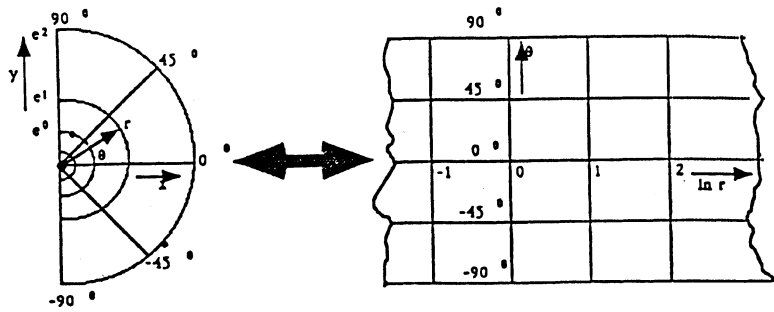


Figure 3. Conformal Mapping of the Cartesian Half-Plane

A Model of the Acquisition of Object Representations in Human 3D Visual Recognition

S. Edelman, D. Weinshall, H. H. Bülthoff, T. Poggio

Center for Biological Information Processing,
MIT E25-201, Cambridge MA 02139, USA

1 Motivation

A common approach to the study of visual recognition postulates that there exist in the visual system representations of familiar objects and scenes. To recognize an object, the system compares it with each of the stored models. Such a comparison would appear possible only after the input image and the stored representations are brought to a common form. Consequently, the nature of representation must be reflected in the performance of the system [7].

One possibility is that the visual system stores a few representative (canonical) views of each known object, along with the information that permits it to normalize the appearance of an input object by computing how it would look like from a canonical viewpoint [9]. Palmer, Rosch and Chase [10] found that canonical views of commonplace objects can be reliably characterized using several criteria. For example, when asked to form a mental image of an object, people usually imagine it as seen from a canonical perspective. In recognition, canonical views are identified more quickly than others, with response times decreasing monotonically with canonicity (as defined, e.g., by subjective ratings).

This dependency of response time on the distance to a canonical view is expected if one draws an analogy between recognition by viewpoint normalization on one hand [6, 18] and mental rotation on the other hand [15, 14]. The very existence of canonical views may then be attributed to a tradeoff between the amount of memory invested in storing object representations and the amount of time that must be spent in viewpoint

normalization. Remembering a frequently encountered view of an object may lead to its faster recognition in subsequent encounters.

By the same argument, no preferred perspective should exist for familiar objects that are equally likely to be seen from any viewpoint. Indeed, there is evidence that normalization effects on recognition latency (as reflected in the existence of preferred views) disappear with practice for a variety of 2D stimuli such as line drawings of common objects [3], random polygons [5], pseudo-characters [4] and stick figures [17].

The aim of the present work is to model some phenomena related to canonical views and viewpoint normalization in object recognition. The work is based on psychophysical experiments that employed methods differing in several respects from previous studies. First, our stimuli were images of novel 3D objects with controlled complexity. This facilitated the study of the effects of object complexity and familiarity on recognition. Second, the stimuli appeared in various 3D orientations, bringing the experimental viewing conditions closer to those of real-world vision. Third, our task did not involve a handedness decision (such as whether the displayed object was a mirror image of the target), avoiding at least one source of criticism in interpreting the results. Fourth, subjects were not required to name the stimuli. This reduced the number of different cognitive modules required for solving the task, bringing the reaction time closer to the actual duration of recognition.

The rest of the paper is organized as follows. The next two sections describe the psychophysical experiments and interpret the results. Sections 4 and 5 outline a simple model of human performance in the experiments and compare simulation results with psychophysical data. Finally, section 6 is a short summary of our conclusions.

2 Experiments

Define the viewpoint coordinates of an observer with respect to an object, θ and ϕ , as the longitude and the latitude of the eye (or the camera) on an imaginary sphere centered at the object. One would expect a function $R(\theta, \phi)$ measuring the ease of recognition for a 3D object possess one or more peaks, corresponding to its canonical views. We assessed the dependency of R on the object's complexity and on its familiarity to the subject, using a two-alternative forced-choice reaction time paradigm.

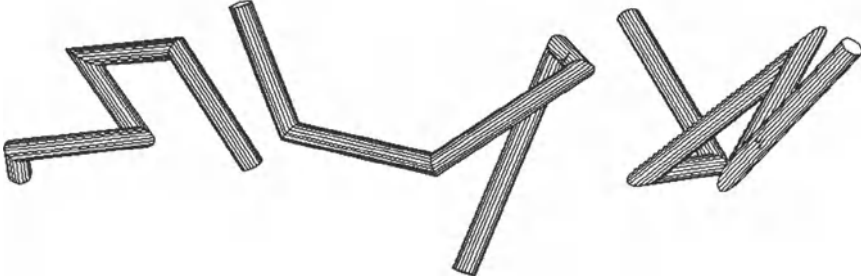


Figure 1: Examples of wire-like objects. Shaded, grey-scale images of similar wires were used as stimuli in the experiments.

2.1 Stimuli

We used the Symbolics S-GeometryTM 3D graphics package to generate novel wire-like objects of small, nonzero thickness (Figure 1). This permitted us to simulate surface shading, while minimizing object self-occlusion. The objects were created in two steps. First, a straight five-segment chain of vertices was made. Second, each vertex was displaced in 3D by a random amount, distributed normally around zero. The variance of the displacements determined the complexity of the resulting wire. Third, the size of the resulting object was scaled, so that all the wires were of the same length.

2.2 Method

Thirty novel 3D objects, generated according to the procedure described above and grouped by average complexity into three sets of ten, served as stimuli in the experiment. 144 evenly spaced images of each of the objects were produced by stepping the camera by 30° increments in latitude and longitude. The images were rendered with the Symbolics S-RenderTM program, using the Lambertian surface reflectance model, with a point light source of intensity 1.0 (located at the camera) and an ambient light source of intensity 0.3. During the experiments, the images were displayed on an CRT monitor, on a dark background, under subdued ambient illumination. The images subtended an angle of approximately 6° at a distance of 120 cm.

The basic experimental run used ten objects of the same complexity as stimuli and consisted of ten blocks, in each of which a different object was defined as the target for recognition. Each block had two phases:

Training: In the beginning of each block, the subject was shown all 144 views of the target twice, in a natural succession. The target was perceived as tumbling in space, with the kinetic depth effect contributing to the three-dimensional appearance of the object.

Testing: In the rest of the block, the subject was presented with a sequence of stimuli, shown one at a time. Half of these were familiar views of the target. The other half were views of the rest of the objects from the current set. For each object, a subset of 16 views (spaced by 90° in latitude and longitude) was used in the test phase. Each of the 16 views of the target appeared during the test phase five times. To facilitate later analysis, the first three and the last two appearances of each view were labeled, respectively, “session 1” and “session 2”.

The appearance of a stimulus was preceded by a fixation point. The stimuli stayed on until the subject responded. The response times were measured in a two-alternative forced choice paradigm. The subject had to press one key if the displayed object was the current target, and another key otherwise. No feedback was given as to the correctness of the response.

Three subjects (the first three authors) participated in the experiment.¹ The basic experiment has been repeated three times (once per complexity group) over a period of a few days. Altogether, 14400 responses were obtained.

To assess the strength of the session effect over a wider range of familiarity, one of the subjects was tested in an additional, identical experiment. This subject saw every view of each target object 10 times, as compared to 5 times for the other two subjects. The results of the two experiments appear below.

2.3 Results

In the following analysis we used only the data from those observations in which the stimulus shown was actually the target (as opposed to one of the distractors)². Latencies of correct responses (response times or RTs) and error rates (ERs) were averaged to yield a single value per session per view per object. RTs longer than 3 sec or shorter than 250 msec were discarded. No evidence of time/accuracy tradeoff was found.

¹The findings reported here have been since replicated with other, naive, subjects [1].

²The reasoning behind this decision, as well as other details of the experiments, can be found in [1].

The decrease of the mean RT with practice was a basic effect that we had expected to find. This effect would have masked any differential effects of familiarity on the recognition of objects from different viewpoints, unless a measure of canonicity (the advantage of some views over others) insensitive to the overall decrease in mean RT were used. We chose the *coefficient of variation* of RT over the different views (defined as the ratio of the standard deviation of RT to the mean of RT) as one measure of the strength of the canonicity effect, and used analysis of variance to find its dependency on familiarity.

A different way to assess the canonical views effect is by looking for an explicit dependency of the RT on the attitude of the object relative to the observer. In this case data cannot be pooled over different objects, unless a common reference attitude is defined. One possibility is to define the (subject-specific) best view for each object as the view with the shortest RT. One could then characterize RT as a function of object attitude by measuring its dependency on $D = D(\text{subject}, \text{target}, \text{view})$, the distance between the best view and the actually shown view. We used regression analysis to characterize $RT(D)$ and $ER(D)$.

The rest of this section summarizes the experimental results (see [1] for details), which are further discussed in section 3.

2.4 Experiment 1: two sessions of 3 and 2 exposures per view

2.4.1 Analysis of response times and error rates

Although the raw results exhibited considerable variation of mean RT and mean ER across subjects, the effects pertinent to the canonical views and mental rotation phenomena were stable and uniform (see below). Mean RTs were 0.75, 0.69 and 0.62 *sec* for low, middle and high complexities. Grouped by session, the RTs were 0.71 and 0.66 *sec* for sessions 1 and 2, respectively³. The only significant interaction was that of complexity $\times$ subject. The mean ER for low complexity was 17.9%, for high complexity – 12.0%, and for middle – 9.7% (the last difference was not significant). The mean ER in session 2, 15.2%, was higher than in session 1, 11.2%.

³All differences among the means reported here and below were found significant by Duncan's multiple-range test at $p < 0.05$, unless otherwise noted.

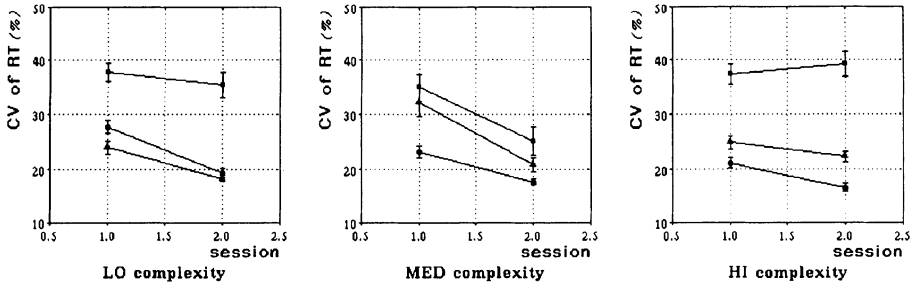


Figure 2: Experiment 1: coefficient of variation of RT (%) over views for the two sessions, by subject and complexity (square, triangle and dot stand for DW, HHB and SYE). The decrease of the c.v. of RT with session is significant.

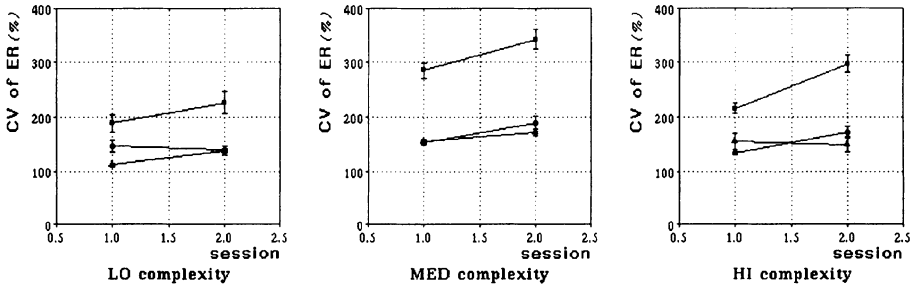


Figure 3: Experiment 1: coefficient of variation of ER (%) over views for the two sessions, by subject and complexity (square, triangle and dot stand for DW, HHB and SYE). The effect of session is significant, mainly due to DW's contribution.

2.4.2 Analysis of the coefficient of variation of response time and error rate

The coefficient of variation of RT over different views of objects decreased with practice (see Figure 2). Effects of subject and session, but not of complexity, were significant. All three means by complexity were close to 26%. The means by session were 29.1% and 23.8% for sessions 1 and 2.

For ER (see Figure 3), all main effects were significant. The means of the coefficient of variation of ER by complexity were 156%, 186% and 206% for low, high and middle sets, respectively (the last difference was not significant). The means by session were 168% and 198% for sessions 1 and 2.

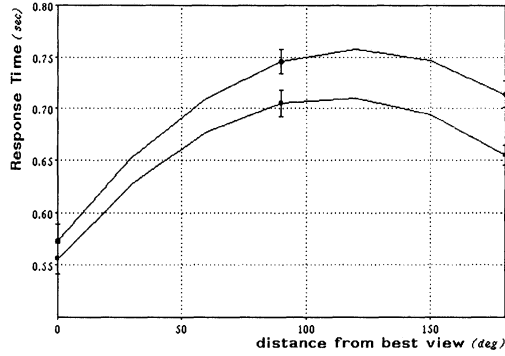


Figure 4: Regression curves of RT on D for the two sessions of experiment 1. Means and standard errors of over 1000 points are shown. RT is measured in *sec*, D – in multiples of 30° . $D = 0$ corresponds to the best view. The lower curve refers to session 2.

2.4.3 Regression analysis of RT, ER

Regression analysis yielded a significant quadratic component. The dependency of RT on D and D^2 for session 1 was $RT = 0.576 + 0.095D - 0.013D^2$. It remained significant for session 2: $RT = 0.558 + 0.076D - 0.010D^2$.

The regression of RT on the distance to a *random* view (fixed for each object and subject), computed as a control, was not significant. Notably, the regression of ER on D and D^2 was also not significant, either for session 1, or for session 2.

The shapes of the regression curves of RT for the two sessions of experiment 1 seem to be different (see Figure 4). A multivariate test of the difference between the two sets of regression coefficients⁴ came short, however, of confirming this impression. This was the main reason for carrying out experiment 2.

2.5 Experiment 2: two sessions of 5 exposures per view each

In this experiment, one of the original subjects (SYE) was tested repeatedly, to elucidate the dependency of regression results on object familiarity (the outcome of an identical experiment, conducted subsequently on four naive subjects, was the same as described

⁴Excluding the intercepts – we were not interested in mere uniform decrease of RT for all views.

below – see [1], experiment 3). For this subject, the responses of both sessions of the previous experiment, consisting together of 5 trials per view per object, were combined, and an additional 5-trial session was performed. The results of this experiment appear below.

2.5.1 Analysis of coefficient of variation of RT, ER

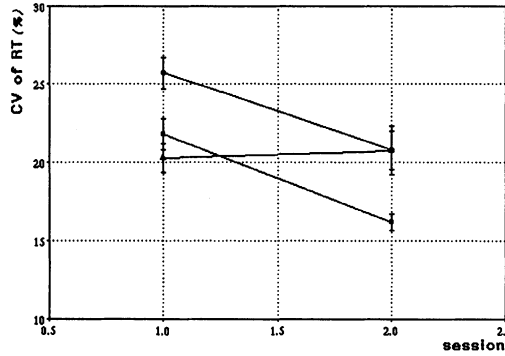


Figure 5: Coefficient of variation of RT over views (%) for the two sessions of experiment 2, by complexity (dot, square and triangle mark low, middle and high complexity, respectively).

The plot of the coefficient of variation of RT for experiment 2 (Figure 5) shows that it decreased with session for the low and the medium, but not for the high, complexity groups. The overall effect of session was significant.

The plot of the coefficient of variation of ER for experiment 2 appears in Figure 6. Only the main effect of complexity was significant. A separate analysis for session by complexity revealed no significant effects of session in any complexity group.

2.5.2 Regression analysis of RT, ER

Regression of RT on D and D^2 for session 1 (see Figure 7) was significant, giving $RT = 0.475 + 0.058D - 0.007D^2$. Importantly, it was not significant for session 2. That is, the dependence of RT on the distance to the best view was strongly diminished. Regression of ER was not significant for both sessions.

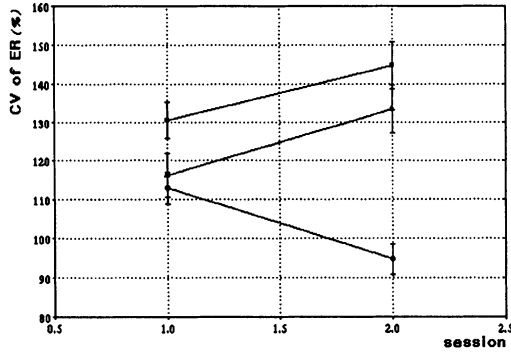


Figure 6: Coefficient of variation of ER rate over views (%) for the two sessions of experiment 2, by complexity (dot, square and triangle mark low, middle and high complexity, respectively).

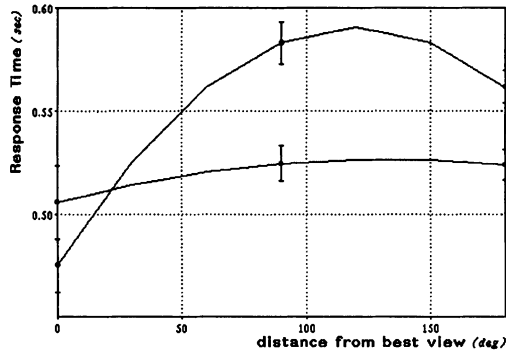


Figure 7: Regression curves of RT on D for the two sessions of experiment 2. Scale labeling is as in the previous regression plot. The flatter curve refers to session 2.

3 Discussion

3.1 Complexity effects

The influence of stimulus complexity on mean RT and ER was in part expected (higher complexity resulted in longer RT and higher ER than middle complexity), and in part unexpected (lower complexity had a similar effect). A possible explanation involves the notion of viewpoint-invariant, non-accidental features of 3D objects [6]. These features are more likely to be present in objects that have, by our definition, higher complexity. While the presence of features such as collinear segments can facilitate recognition, having too many of them would have an opposite effect, e.g., by prompting the subject to resort to a more complicated procedure. Having too few of these features could also impede recognition (by increasing ambiguity).

Stimulus complexity had no effect on the coefficient of variation of RT over views. It appears that most of the variation of RT (as opposed to the mean RT) is due to factors other than complexity, such as the general outlook of our stimuli (e.g., an elongated wire seen end-on would be naturally harder to recognize than the same wire seen from the side). On the other hand, stimulus complexity affected the coefficient of variation of ER over views. We do not attempt to interpret this effect, because of the possible subject $\times$ complexity interaction (see the difference between the data for subject DW and the other two subjects in Figure 3).

3.2 Session (familiarity) effects

Our data indicate a clear effect of familiarity on the prominence of canonical views, at least for the kind of objects we have used as stimuli. Familiarity appears to reduce the differences in RT among different views of the object (see Figure 2), and to render insignificant possible effects of mental rotation, as manifested in the dependency of RT on the distance to the canonical view (Figure 7). We interpret session effects in the absence of feedback as an indication of *imprinting* of familiar views that happens merely as a result of repeated exposure.

3.3 Interpreting regression results

Experimental results in which recognition time of an object depended on the amount of rotation necessary to bring it to a familiar orientation have been previously interpreted

in terms of mental rotation [17]. The major argument in favor of this interpretation is indirect and has to do with similarity between the slope of the regression curve in recognition and in classical mental rotation tasks [16, 14]. The reciprocal of the coefficient of D in the regression equation for $RT(D)$ in session 1 in our experiments (approximately 300 deg/sec) is also consistent with that of mental rotation. This result, along with the apparent absence of an orderly dependence of ER on D , can be accommodated by a theory of recognition that involves two distinct stages: normalization and comparison (cf. Ullman's recognition by alignment [18]). In the normalization stage, the image and a model are brought to a common attitude in a visual buffer. This operation could be done by a process analogous to mental rotation, which would take time proportional to the attitude difference between the image and the model. Subsequently, a comparison would be made between the two. The time to perform the comparison could depend, e.g., on the object's complexity, but not on its attitude, so that the comparison stage would contribute a constant amount to the overall recognition time. On the other hand, the error rate of recognition would be largely determined by the comparison stage. With practice, more views of the stimuli could be retained by the visual system, resulting in a smaller average amount of rotation necessary to normalize the input to a standard, or canonical, appearance. Thus, the mean response time (determined by the normalization process) would decrease, but the mean error rate (determined by the comparison process) would not, because of the absence of feedback to the subject. This is compatible with our observations.

The strong quadratic component in the regression equations for $RT(D)$ may signify the presence of more than one preferred, or canonical, view. Imagine the viewing sphere (see section 2) centered around a wire-like object, with the best (shortest-RT) view at the north pole. Then the view from the south pole of the sphere (at $D = 180^\circ$ from the north pole) ought to yield shorter RT than views from the equator, because the projection of a wire looks almost the same from two diametrically opposite directions. This may explain the shape of the regression curve for $RT(D)$.

3.4 Summary of recognition psychophysics

To recapitulate, our main findings are as follows.

- Stimulus complexity has little effect on the variation of RT over views;
- Stimulus familiarity reduces the variation of RT over views;

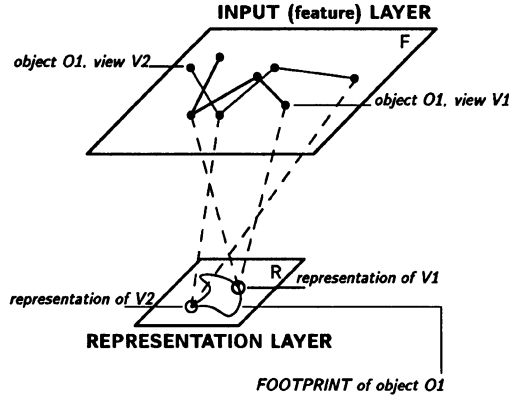


Figure 8: The network consists of two layers, F (input, or feature, layer) and R (representation layer). Only a small part of the projections from F to R are shown. The network encodes input patterns by making units in the R-layer respond selectively to conjunctions of features localized in the F-layer. The curve connecting the representations of the different views of the same object in R-layer symbolizes the association that builds up between these views as a result of practice.

- Familiarity reduces the effect that can be interpreted in terms of mental rotation, namely, the dependency of RT on the distance to the canonical view.

These effects support the notion of a tradeoff between time required for viewpoint normalization and memory invested in storing multiple views of objects. One possible computational interpretation of our findings is in terms of a two-stage process of recognition by normalization, gradually superseded with practice by a more memory-intensive, less time-consuming strategy. The following section explores a different model, which appears to be equally capable of reproducing our data.

4 The model

Can a simpler computational process than the two-stage recognition by alignment account for our results? To address this question, we simulated the experiments described above with a two-layer network of thresholded summation units (see Figure 8) [2]. The “stimuli” in the simulation were the projections of the vertices of the same wire objects used in the actual experiment. This allowed us to make a direct qualitative comparison of

the simulation results with the data from human subjects. In the following, we describe briefly the principle of operation of the network.

4.1 Learning

The first (input, or feature) layer of the network is a feature map. Every unit in the F-layer is connected to all units in the second (representation) layer. The strength of these “vertical” (V) connections has initially a Gaussian distribution. In addition, the units in the representation layer are connected among themselves by lateral (L) connections, whose initial strength is zero. The V-connections form specific-view representations. The L-connections form associations among different views of the same object.

The input is a sequence of appearances of an object, encoded by the 2D locations of concrete sensory features (line terminators and corners) rather than a list of abstract features. At the first presentation of a stimulus several representation units are active, all with different strengths (due to the Gaussian distribution of vertical connection strengths). We employ a winner-take-all (WTA) mechanism to identify the strongest active R-unit. Hebbian relaxation then enhances V-connections from the input layer to the winner. Specifically, the connection strength v_{ab} from F-unit a to R-unit b changes by

$$\Delta v_{ab} = \min \{ \alpha A_a [A_b - T_b] v_{ab}, v^{max} - v_{ab} \} \cdot \frac{v^{max} - v_{ab}}{v^{max}} \quad (1)$$

where A_u and T_u are the activity and the threshold of the unit u , v^{max} is an upper bound on connection strength and α is a parameter controlling the rate of convergence ($[x]$ is defined as $\max\{x, 0\}$). The threshold of the winner R-unit is increased by

$$\Delta T_b = \delta \sum_a \Delta v_{ab} A_a \quad (2)$$

where $\delta \leq 1$. As a result, this R-unit encodes the spatial structure of a specific view, responding selectively to this view after only a few (two or three) presentations.

The principle by which specific views of the same object are grouped is that of temporal association. New views of the object appear in a natural order, corresponding to their succession during an arbitrary rotation of the object. The lateral (L) connections in the representation layer are modified by a time-delay Hebbian relaxation. L-connection w_{bc} between R-units b and c that represent successive views are enhanced in proportion to their closeness in time, up to a certain time difference K :

$$\Delta w_{bc} = \sum_{|k| < K} \gamma_k [A_b^t - T_b^t] [A_c^{t+k} - T_c^{t+k}] \cdot \frac{w^{max} - w_{bc}}{w^{max}} \quad (3)$$

The appearance of a new object is explicitly signalled, so that two different objects do not become associated by this mechanism. The parameter γ_k decreases with $|k|$ so that the association is stronger for units whose activation is closer in time. In this manner, a *footprint* of temporally associated view-specific representations is formed in the second layer for each object. Together, they form a distributed multiple-view representation of the object.

4.2 Recognition

An input presented to the feature layer produces a pattern of activity in the representation layer. We define the object whose footprint is closest to this activity pattern as the outcome of recognition. A measure of closeness between two patterns is provided by *correlation*. This choice may be clarified by considering a model of decision-making in recognition in which many units (possibly with different initial levels of activation) encode the known entities (one unit per entity; cf. [8, 11]. In our case several units together encode an object.). When an input is present, each unit's activation is increased in proportion to the similarity between the input and the concept that the unit represents. The decision threshold, initially kept high to discourage false alarms, is gradually decreased, until it is exceeded by some unit's activation. Recognition latency in this scheme clearly depends on the activation induced by the input in the would-be strongest representation unit. In our scheme, this activation is measured by the correlation between the actual footprint induced by the input and the prototypical memory trace of this footprint. This correlation also serves as an analog of response time.

5 Simulation

We were able to reproduce all three main results of the psychophysical experiment described in the previous section, with a random choice of the parameters of the network model. First, no dependency of the coefficient of variation of correlation (CORR) over views on stimulus complexity was found. Second, the variation of CORR over views decreased with practice (Figure 9). Third, the dependence of CORR on stimulus attitude (Figure 10) diminished with practice.

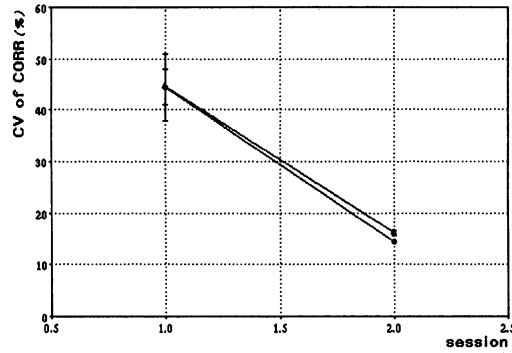


Figure 9: Coefficient of variation of CORR over views for the two sessions, by complexity, before the introduction of shortcuts into the footprint (see text).

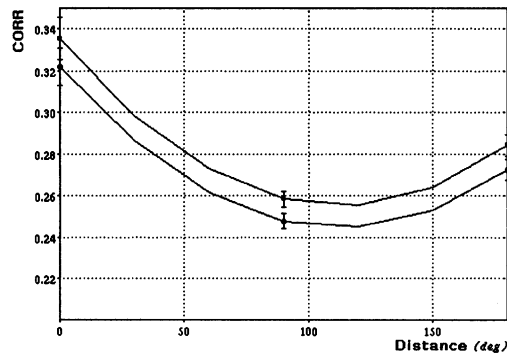


Figure 10: Regression of CORR on distance to the best view, by session, before the introduction of shortcuts into the footprint (see text). Note the similarity to the regression plot of experiment 1, keeping in mind that high CORR is analogous to low RT.

The effect of session on the coefficient of variation of CORR was significant ($F(1, 16) = 15.88, p < 0.001$). A multivariate test of the difference between the sets of regression coefficients corresponding to sessions 1 and 2 (excluding the intercept) was insignificant ($F(2, 157) = 1.5, p = 0.23$; compare this to the outcome of an analogous test of the regression coefficients for experiment 1 in section 2.4.3), raising questions regarding the ability of the model to replicate the flattening out of the regression of RT on D . To further test this ability, we allowed the enhancement of the lateral connections in the representation layer during the test phase of the simulated experiment in addition to their enhancement during the training phase (controlled by γ_k in equation 3). As a result, shortcuts appeared in the sequences of R-units representing successive views of objects, obliterating the linear structure of these sequences (footprints), responsible for the semblance of mental rotation apparent in Figure 10.

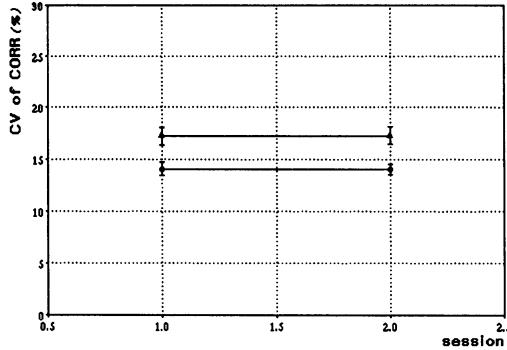


Figure 11: Coefficient of variation of CORR over views for the two sessions, by complexity, after the introduction of shortcuts into the footprint (see text).

Introducing the shortcuts enhanced the session effect, increasing the significance of the difference between the regression coefficients of CORR on D for the two sessions ($F(2, 157) = 2.6, p < 0.08$; see Figure 12). The effect of shortcuts on the coefficient of variation of CORR was even stronger (compare Figure 11 with Figure 9). Apparently, already the first session caused the CORR characteristics for the different views to reach their steady-state values.

Finally, we assessed the generalization ability of the network to novel views of familiar objects. Rock [12, 13] found that people have difficulties in recognizing or imagining wire-

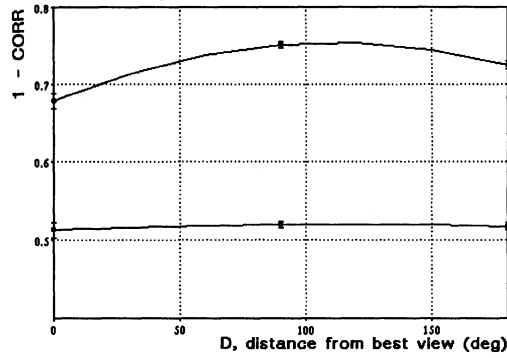


Figure 12: Regression of $CORR$ on distance to the best view, by session, after the introduction of shortcuts into the footprint (see text). To facilitate comparison with human subject data (Figure 7), $1 - CORR$ rather than $CORR$ is plotted against the distance to the best view. Note the flattening of the regression curve in session 2.

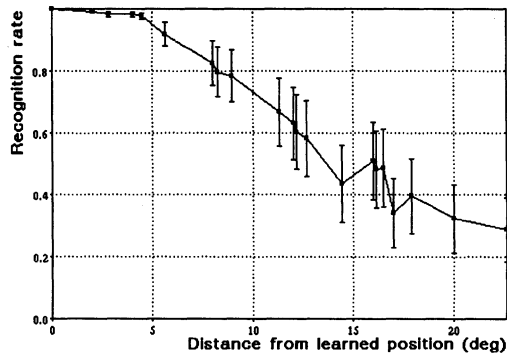


Figure 13: Performance of the net on novel orientations of familiar objects (mean of 10 objects, bars denote the variance).

frame objects in a novel orientation that differs by more than 30° from a familiar one. We have tested the network model on views obtained by rotating the objects away from learned views by as much as 24° (see Figure 13). The classification rate was better than chance for the entire range of rotation, but decreased to 35% at 24° (chance level was 10% for ten objects). This result establishes another parallel between human performance and the performance of our model in a recognition task.

6 Summary

Two main effects are apparent in our data: (1) the decrease with practice in the variation of recognition latency over different views of an object and (2) the disappearance of the dependence of the latency on the object's orientation relative to a canonical view. Both these effects support the notion of a tradeoff between time required for viewpoint normalization and memory invested in storing multiple views of objects. A standard interpretation (questioned by some researchers [13]) of the second effect is in terms of mental rotation (e.g., for the purpose of alignment) of object representations, that becomes unnecessary when many specific views of objects are stored as a result of practice.

The simulated replication of our psychophysical data by a model that has no a priori mechanism for "rotating" object representations indicates that a different interpretation of findings that are usually taken to signify mental rotation is possible. Cooper ([14], p.160ff) opened the way to such interpretation: "one-to-one correspondence between the intermediate states in a mental rotation and a rotation of an external object need not be one of a structural isomorphism between the internal representation undergoing mental rotation and the external object undergoing the physical rotation". The footprints formed in the representation layer in our model provide a hint as to what the substrate upon which the mental rotation phenomena are based may look like.

Acknowledgements

We thank Jeremy Wolfe and Ellen Hildreth for their comments on an early draft of this paper. This paper describes research done within the Center for Biological Information Processing in the MIT Department of Brain and Cognitive Sciences. The Center's research is sponsored by grants from the Office of Naval Research (ONR), Cognitive and Neural Sciences Division; by the Alfred P. Sloan Foundation; and by the National

Science Foundation. TP is supported by the Uncas and Helen Whitaker Chair at the MIT Whitaker College, by the MIT Artificial Intelligence Laboratory, by Hughes Aircraft Corporation, and by the NATO Scientific Affairs Division. SE and DW are supported by Chaim Weizmann Postdoctoral Fellowships and by grants from the National Science Foundation. HHB is currently with the Department of Cognitive and Linguistic Sciences, Brown University.

References

- [1] S. Edelman, H. Bülthoff, and D. Weinshall. Stimulus familiarity determines recognition strategy for novel 3D objects. A.I. Memo No. 1138, AI Laboratory, MIT, July 1989.
- [2] S. Edelman and D. Weinshall. A self-organizing multiple-view representation of 3D objects. A.I. Memo No. 1146, AI Laboratory, MIT, August 1989.
- [3] P. Jolicoeur. The time to name disoriented objects. *Memory and Cognition*, 13:289–303, 1985.
- [4] A. Koriat and J. Norman. Mental rotation and visual familiarity. *Perception and Psychophysics*, 37:429–439, 1985.
- [5] A. Larsen. Pattern matching: effects of size ratio, angular difference in orientation and familiarity. *Perception and Psychophysics*, 38:63–68, 1985.
- [6] D. G. Lowe. *Perceptual organization and visual recognition*. Kluwer Academic Publishers, Boston, MA, 1986.
- [7] D. Marr. *Vision*. W. H. Freeman, San Francisco, CA, 1982.
- [8] J. Morton. Interaction of information in word recognition. *Psychological Review*, 76:165–178, 1969.
- [9] S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. In J. Long and A. Baddeley, editors, *Attention and Performance IX*, pages 135–151. Erlbaum, Hillsdale, NJ, 1981.
- [10] S. E. Palmer. The psychology of perceptual organization: a transformational approach. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and machine vision*, pages 269–340. Academic Press, New York, 1983.
- [11] R. Ratcliff. Parallel processing mechanisms and processing of organized information in human memory. In J. A. Anderson and G. E. Hinton, editors, *Parallel models of associative memory*. Erlbaum, Hillsdale, NJ, 1981.
- [12] I. Rock and J. DiVita. A case of viewer-centered object perception. *Cognitive Psychology*, 19:280–293, 1987.
- [13] I. Rock, D. Wheeler, and L. Tudor. Can we imagine how objects look from other viewpoints? *Cognitive Psychology*, 21:185–210, 1989.
- [14] R. N. Shepard and L. A. Cooper. *Mental images and their transformations*. MIT Press, Cambridge, MA, 1982.

- [15] R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171:701–703, 1971.
- [16] S. Shepard and D. Metzler. Mental rotation: effects of dimensionality of objects and type of task. *J. Exp. Psychol.: Human Perception and Performance*, 14:3–11, 1988.
- [17] M. Tarr and S. Pinker. Mental rotation and orientation-dependence in shape recognition. *Cognitive Psychology*, 21, 1989.
- [18] S. Ullman. Aligning pictorial descriptions: an approach to object recognition. *Cognition*, 32:193–254, 1989.

Part 2

Hands and Tactile Perception

The Perception of Mechanical Stimuli Through the Skin of the Hand and its Physiological Bases

R.T. Verrillo and S.J. Bolanowski, Jr.
Institute for Sensory Research
Syracuse University
Syracuse, NY 13244, U.S.A.

Abstract

A series of experiments was performed in which psychophysical measurements were made at threshold and suprathreshold levels of stimulation under carefully controlled laboratory conditions. A wide range of stimulus parameters was explored including sinusoidal frequency, intensity, size of contactor, surround condition, skin temperature, and body site. These results are compared to experiments in which responses to sinusoidal displacements were measured from receptor nerve fibers in this and other laboratories. Our work has culminated in a model of cutaneous mechanoreception that proposes four discrete information channels that combine at threshold and suprathreshold levels to signal tactile perception. The psychophysical channels and their physiological counterparts are:

- 1) the Pacinian channel, mediated by Pacinian-corporuscle afferents;
- 2) the Non-Pacinian I channel, mediated by rapidly-adapting afferents (Meissner corpuscles);
- 3) the Non-Pacinian II channel, mediated by slowly-adapting Type II afferents (Ruffini endings); and
- 4) the Non-Pacinian III channel, mediated by slowly-adapting Type I afferents (Merkel cell-neurite complex). These channels operate over specific bands of vibratory frequencies and the channels partially overlap in their absolute sensitivities.

Introduction

All sensory systems can be divided into three major components. One of these is the receptor surface, which is responsible for the transduction of physical energy into neural signals. An example of this is the retina where the rods and cones together with the bipolar, horizontal, and amacrine cells transduce and condition the light energy impinging upon the eye. The second element is the transmission line that sends information from the receptor surface to processing areas located more centrally. For vision, this part is considered to be the ganglion-cell axonal processes which form the optic nerve. The third component is comprised of the central processing centers which decipher, modify, and integrate the sensory information. This is typified by the lateral geniculate nucleus and all of the visual

regions of the cortex. Of course, other schemes can be used to subdivide any sensory system. For example, each system can be disassembled along operational and functional lines rather than by anatomical structures. Sensory systems such as vision can be divided into psychophysically or physiologically defined "channels" which funnel the information centrally. One example of this is the chromatic and luminance channels; another is the separation of the spatial- and temporal-frequency channels.

Unlike the eye, which transduces light energy exclusively, the skin contains receptors that mediate the perception of a variety of experiences produced by several forms of energy: mechanical (vibration and pain), thermal (warmth and cold), and chemical (pain). This receptor surface can be divided along anatomical as well as functional lines with the various subsystems each possessing its own, partially independent, organization. For example, the mechanical (i.e., vibratory) aspects of touch, at least for glabrous regions of skin, which are hairless, utilizes at least four anatomically distinct mechanoreceptor types, namely Pacinian corpuscles, Meissner corpuscles, Ruffini endings, and Merkel-cell neurite complexes (Vallbo and Johansson, 1986; Bolanowski, Gescheider, Verrillo and Checkosky, 1988). The output of each is transmitted centrally via the peripheral nerves called Pacinian-corpuscle (PC) nerve fibers, Rapidly Adapting (RA) fibers, Slowly Adapting type-I (SA-I) fibers and Slowly Adapting type-II (SA-II) fibers. The impulses over these fibers are transmitted to spinal-cord centers as well as to the higher regions of the central nervous system such as the dorsal column nuclei, thalamus, and cortex. We (Bolanowski, et al., 1988) have recently proposed that each of the four peripheral pathways of this subsystem may be mapped operationally onto four psychophysically defined channels. Thus, the perception of the mechanical aspects of touch in all likelihood is comprised of a mixture of information originating from all four channels.

The purpose of this paper is to show in a generic way how mechanical stimuli are transduced, how the information is encoded, and how the information may combine from the four psychophysically distinct channels to form perceptions of the mechanical aspects of touch. For this purpose we will focus on the Pacinian corpuscle, the tactile mechanoreceptor best understood and perhaps prototypical of all tactile mechanoreceptors. We hope to show how this and the other tactile mechanoreceptors can be linked to the four psychophysically-defined sensory channels of touch.

The Pacinian Corpuscle

The Pacinian corpuscle is a blimp-shaped, capsular receptor found in the lower dermis and subcutaneous tissue of all mammals. (See, for example, Cauna and Mannan, 1958; Poláček and Mazanec, 1966; Nishi, et al., 1969). It is the largest (1.0 mm by 0.5 mm) mechanoreceptor in the skin and is composed of an outer capsule formed by lamellar cells which concentrically surround an inner core region (Chouchkov, 1971). The inner core is formed by hemilamellae which compress the nerve fiber innervating the capsule (Quilliam and Sato, 1955). The myelinated, single nerve fiber enters the corpuscle through one pole of the capsule. Upon reaching the inner core, the fiber loses its myelin sheath and becomes elliptical in cross section sandwiched between the hemilamellar cells of the inner core. The unmyelinated region which extends the entire length of the corpuscle is the region where mechanotransduction takes place. A complete description of the ultrastructural anatomy of the Pacinian corpuscle can be found in Spencer and Schaumberg (1973). We have focussed our work primarily on the Pacinian corpuscle since it is prototypical in its response properties. In fact, the current position of many tactile investigators is that the principles of transduction are the same in all tactile mechanoreceptors except that the accessory structures impose a mechanical filter which determines the frequency response of each receptor type. This is directly analogous to the chromophore structure of opsin in the retinal receptors called rods: rhodopsin having an action spectrum in the 460-540 nm range. Other photopigments (e.g., iodopsin) impart the spectral sensitivity of the cone-receptor types. Other factors that can contribute to the frequency response are response criteria and the intervening tissues between the receptor surface and the outside world. For example, in vision, the lens and ocular media filter various wavelengths of light, particularly in the ultraviolet region. For the vibratory stimuli that are used in our laboratory, the skin plays no such role. Van Doren (1989) has shown that the skin is functionally transparent for stimuli in the temporal domain, although it plays a major role in the spatial domain.

Stimulus deformations produce receptor potentials in the unmyelinated portion of the nerve fiber: if they are of sufficient amplitude a neural impulse will be generated and passed along the myelinated fiber to the central nervous system. Figure 1 shows the relationship among a stimulus, the receptor potential, and the action potential generated by the underlying receptor potential for a single, isolated Pacinian corpuscle. In both the physiological and psychophysical experiments to be described, sinusoidal bursts of displacements were used as the preferred stimulus because the use of sinusoids permits a systems-analysis approach to the problem. The power of this approach has been amply demonstrated in the excellent work performed on both the auditory and visual systems. The lowest trace in Fig. 1 signifies two

cycles of the stimulus. For low stimulus intensities (Fig. 1A) only a barely discernible receptor potential can be seen in response to the stimulus. At moderate stimulus intensities (Fig. 1B) the corpuscle fires one impulse for every cycle (1:1) of stimulation. At more

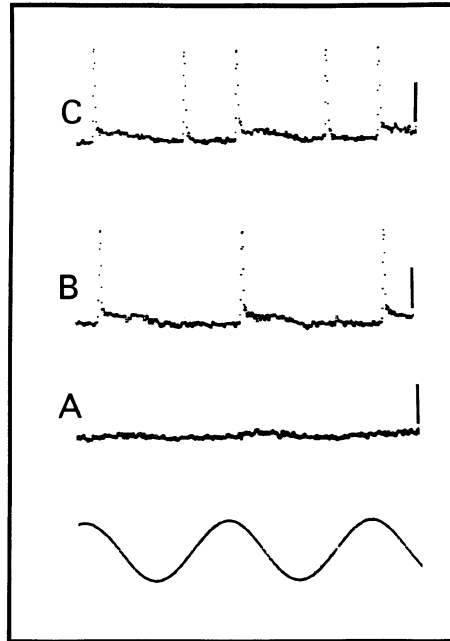


Fig. 1 Relationship among stimulus, receptor potential, and nerve impulses of a Pacinian corpuscle. Shown is a voltage trace signifying a 50-Hz vibratory stimulus (Bottom trace) and the response of a corpuscle at three different stimulus intensities (A,B,C). The intensity of stimulation increased from A to C. In the stimulus trace, a negative deflection signifies compression of the corpuscle. A positive deflection in the response traces of A through C corresponds to depolarization.

intense stimuli, a 2 spike-per-cycle (i.e., 2:1) firing pattern occurs. A typical relationship between stimulus intensity and impulse-firing rate is shown in Fig. 2. The firing rate in impulses per second is plotted as a function of stimulus displacement (NOTE: In this and similar figures to follow, stimulus amplitude is plotted in dB re 1 μ m peak.) At lower stimulus amplitudes (A), the rate-intensity characteristic rises steeply with increases in stimulus intensity. Plateaus occur in the rate-intensity characteristic at multiples (C) and submultiples (B) of the stimulus frequency, a phenomenon known as phase locking. At moderate stimulus displacements, the corpuscles produce a 1:1 firing pattern over a wide range of stimulus amplitudes C-D). It is the 1:1 firing pattern that has been suggested as the

neural code used by the nervous system to signal activation of Pacinian corpuscles. However, we now know that the code is probably much more complicated (see below). At sufficiently high stimulus intensities (E) the corpuscle can fire in a 2:1 manner. The mechanism responsible for producing the rate-intensity functions and the characteristic phase-locking patterns is the underlying mechanotransductive process revealed by the

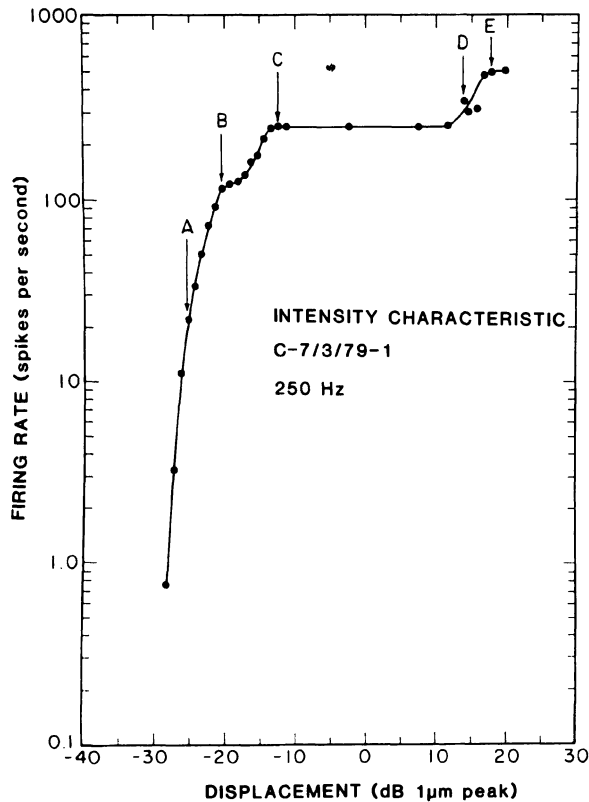


Fig. 2 Rate-intensity characteristics obtained from the corpuscle in response to a 250 Hz stimulus.

receptor potential. The receptor-potential response is basically an analog conversion of the mechanical stimulus. Impulse initiation produced by the receptor potential is simply an analog-to-digital conversion. A typical receptor-potential response to a large (50 μm peak) sinusoidal (50 Hz) displacement is shown in the inset of Figure 3. The lower trace of the inset shows the stimulus and the upper trace is the receptor potential. Notice that the receptor potential is highly nonlinear, possessing the properties of frequency doubling and hysteresis. These phenomenon can best be appreciated by plotting the stimulus amplitude versus the

response in the form of a Lissajous pattern. The function so generated is shown in Fig. 3 and is traditionally referred to as an input-output (I-O) function. Positive displacements represent stimulus compression of the corpuscle and negative values are stimulus withdrawals. The stimulating probe is statically indented $110\text{ }\mu\text{m}$ into the receptor before being activated in order to ensure that the probe never fully withdraws from the corpuscle. The I-O function so obtained is non-monotonic, displaying a roughly linear response to stimulus compression, with a reversal of the response as seen in the third quadrant of the I-O plot. This phenomenon can be modeled by a system that produces a response somewhere between half- and full-wave rectification. Since the rectification is not symmetrical across the zero-displacement

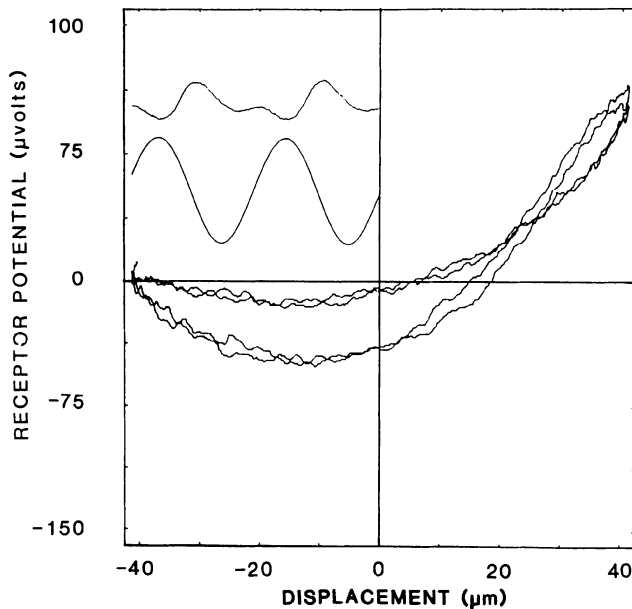


Fig. 3 The input-output transfer function relating the receptor-potential amplitude to the stimulus. The amplitude of the 100-Hz displacement was $40\text{ }\mu\text{m}$ peak. Static indentation was $110\text{ }\mu\text{m}$. The inset shows the actual relationship between the stimulus (bottom trace) and the receptor-potential response. In the generation of the I-O response, the phase relationship between the response and stimulus was changed from that shown in the inset so that the peak receptor-potential response was referenced to maximum displacement.

axis, we have termed the receptor-potential I-O function an *asymmetric full-wave rectification*. The I-O functions indicate that at small stimulus displacements, the receptor potential will be linear (i.e., small transitions along the displacement axis). The larger the stimulus displacements, the more non-linear will be the receptor-potential response. Impulse

initiation in the nerve fiber occurs when the transmembrane potential (in this case the receptor potential) reaches a particular depolarization level in a given amount of time. Since the I-O function of Pacinian corpuscles is asymmetric, the positive-going displacements generate greater receptor-potential amplitudes than the negative-going, given a fixed displacement level. The receptor-potential depolarization in the first quadrant of the I-O function is responsible for the low-intensity limb of the rate-intensity function and the 1:1 plateau. The 2:1 firing pattern is produced when the stimuli are large enough to produce the second depolarization, evidenced by the asymmetric full-wave rectification properties in the third quadrant of the I-O function.

Another important characteristic of systems in general is the frequency response. Traditionally, the frequency response is obtained by holding the input level constant, varying the stimulus frequency, and measuring the output. In sensory systems it is much easier to hold the output constant, as we do in measuring psychophysical detection thresholds (see below). For the physiological results, these functions can be generated using

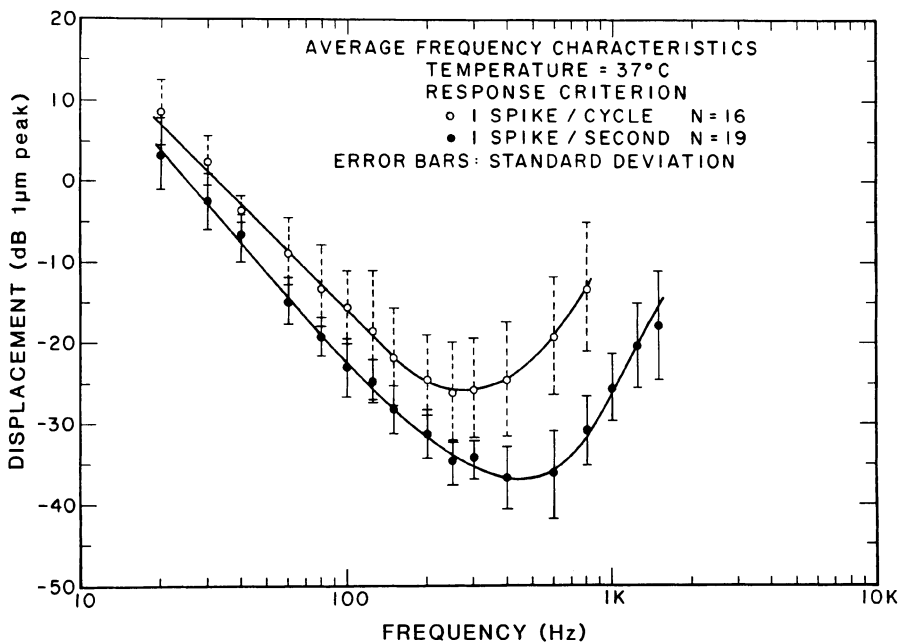


Fig. 4 Average frequency characteristics of Pacinian corpuscles. The error bars signify the standard deviations. The response criterion and number of corpuscles used in the averaging was: ○, 1 spike/cycle, n=16; ●, 1 spike/sec, n=19.

receptor-potential responses or neural-impulse firing rates. The experience of a "sensation" requires that the information transduced at the receptor be transmitted to more-centrally located processing regions. For this purpose, all sensory systems use neural impulses as the digital code which is passed over the transmission portion of the system. Since one of the goals of our laboratory is to link physiology to psychophysics, we will focus on frequency characteristics obtained when using neural-impulse information. Figure 4 shows the average frequency response obtained on many corpuscles using two response criteria: the first, 1 impulse-per-stimulus burst and the other 1 impulse-per-stimulus cycle (1:1). The characteristics are U shaped with a maximal sensitivity around 250 Hz. The figure shows that the Pacinian corpuscle is responsive primarily to vibratory stimuli in the range of 40 Hz to 1 KHz. The significance of this will become clear in the psychophysical portion of this report.

The particular code that the tactile or any other sensory system uses to signal a sensory event is not known at this time, but there are several possibilities. These include, but are not limited to: firing rate (e.g., impulses/sec); number of impulses per stimulus; number of

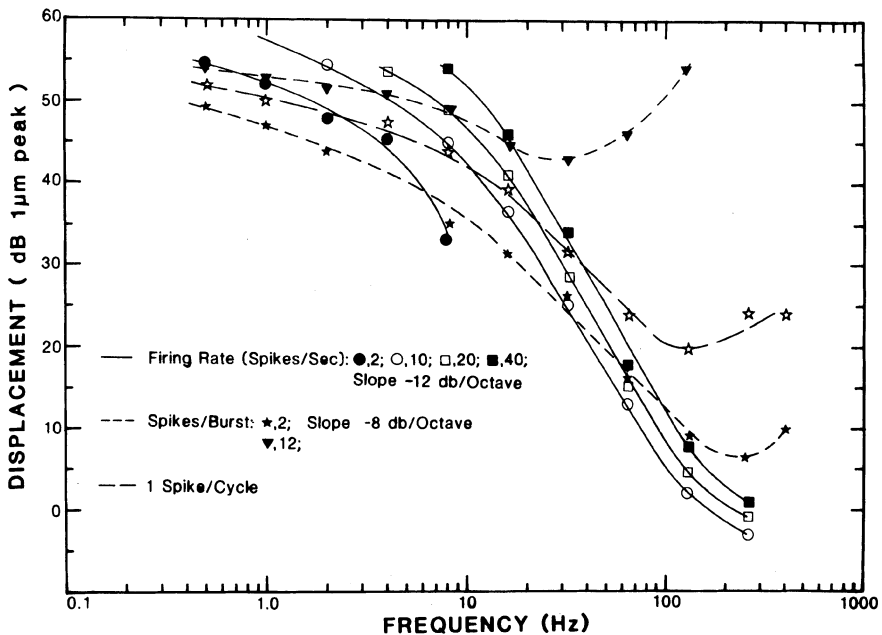


Fig. 5 Frequency characteristics of Pacinian corpuscles. Data replotted from Johansson, et al. (1982). The characteristics were generated by first extrapolating and interpolating their equi-amplitude response profiles and plotting the results as rate-intensity characteristics. The frequency responses shown here were then constructed from the rate-intensity characteristics by choosing various response criteria as shown in the figure and plotting the required stimulus amplitude versus stimulus frequency to attain the required neural response.

impulses per cycle of stimulation; and, changes in the stochastic properties of firing, especially when there is significant noise (spontaneous activity) in the system. When we try to link psychophysical behavior to underlying physiological mechanisms, we must assume that the criterion is related to some aspect of the neural activity. This is somewhat problematical, since different criteria can produce vastly different threshold-frequency characteristics. This is demonstrated clearly in Fig. 5, the data of which have been replotted from Johansson, Landström and Lundström (1982). They obtained responses from human Pacinian-corporcle fibers (as well as the other fiber types) in response to vibratory stimuli by using the percutaneous electrophysiological technique pioneered by Vallbo and Hagbarth (1968). Johansson, et al. (1982) plotted their results as equi-intensity contours and we have replotted their Pacinian-corporcle data using three fixed criteria to demonstrate the effects of response criteria on the frequency characteristics. Figure 5 shows that the frequency response changes dramatically when different criteria are used. Similar plots of the other three fiber types were also made and they show the same general effects. The criteria that we selected for analysis were based on a number of considerations including factors such as temporal summation, spatial summation, temperature, response characteristics of single fibers, and others as outlined in detail in Bolanowski, et al. (1988). Before the relation between the physiological and psychophysical aspects of touch can be established, it is first necessary to describe the psychophysical methods used and the research results obtained that have allowed us to define the capabilities of the tactile system.

Psychophysical Experiments

Although four fiber types have been identified in man, until recently it had not been shown that all contribute to the sensation of touch. Indeed, about 30 years ago we were firmly convinced that all sensations originating in the skin were produced by the stimulation of one type of receptor; a single morphological entity that was excited by vibration, pressure, heat, and cold, or physical damage. The sensations that we experience were thought to be the product of the spatiotemporal patterning of impulses produced by this ubiquitous but unidentified receptor that was able to transduce mechanical, thermal, and noxious energies (Sinclair, 1967).

We can leap over many years of painstaking research by stating simply that when we (Verrillo, 1963) measured vibratory detection thresholds as a function of frequency, the function had two limbs; one that was relatively flat at low frequencies, and another that was U shaped between 40 and 1,000 Hz. What was important for the success of those early

experiments was the careful definition of the stimulus. The experimental approach has been further refined since then, but the basic techniques are still the same: 1) the use of sinusoidal vibratory displacements applied to the thenar eminence of the right hand by a circular contacting surface (contactor) which can be varied in size (0.008 to 2.9 cm²); 2) the contactors are surrounded by a rigid surface, separated from the contactor by a 1-mm gap, that confines the deformations to the area of stimulation; 3) displacements are produced around a static indentation of 0.5 mm into the skin to ensure contact between the skin and the contactor during the sinusoidal excursions; 4) regardless of the duration of the stimulus which can range from 10 -2400 ms depending upon the experiment, appropriate rise-fall times are used to avoid onset/offset transients; and 5) in order to reduce bias in the response of the observer thresholds are measured on human observers using a two-alternative, forced-choice tracking procedure (Zwislocki, Maire, Feldman, and Rubin, 1958) designed for this purpose. The recent addition of maintaining skin-surface temperature to within $\pm 0.5^{\circ}\text{C}$ by circulating water through hollow chambers in the stimulus apparatus has provided an even better definition of the stimulus.

Using the techniques described above and testing thresholds to very low (0.4 Hz) vibratory frequencies, we have recently shown that the psychophysical threshold characteristic as measured at the thenar eminence of the hand is actually composed of three different portions (Bolanowski, et al., 1988). The characteristic is given by the solid circles in Fig. 6 and was obtained using a large contactor surface (2.9 cm²) with the skin-surface temperature maintained at 30 °C.

The low-frequency portion of the curve extends from 0.4 to 3.0 Hz and appears to be insensitive to changes in frequency. A second portion is frequency dependent with an approximate slope of -5 dB/octave between 3.0 and 40 Hz. The third portion is U shaped from 40 to 500 Hz with a slope of -12 dB/octave in the lower frequencies and maximally sensitive in the region of 250 to 300 Hz.

The receptor system responsible for the high-frequency, U-shaped limb of the curve is highly sensitive, frequency dependent, and capable of integrating energy over time and space (Verrillo, 1962, 1963, 1965, 1966, 1968; and others). Because of its close similarity to physiological curves in shape, sensitivity, and position, obtained independently in many laboratories, we identified this system as the P-channel. An extensive series of experiments, including masking, adaptation, and matching procedures, as well as manipulations of stimulus size and duration, confirmed that the U-shaped curve was indeed due to the activation of Pacinian corpuscles. The short and long dashed line shows the shape and

location of the P channel as measured by many experiments (see Verrillo, above and Bolanowski and Verrillo, 1982; Gescheider, Frisina and Verrillo, 1979).

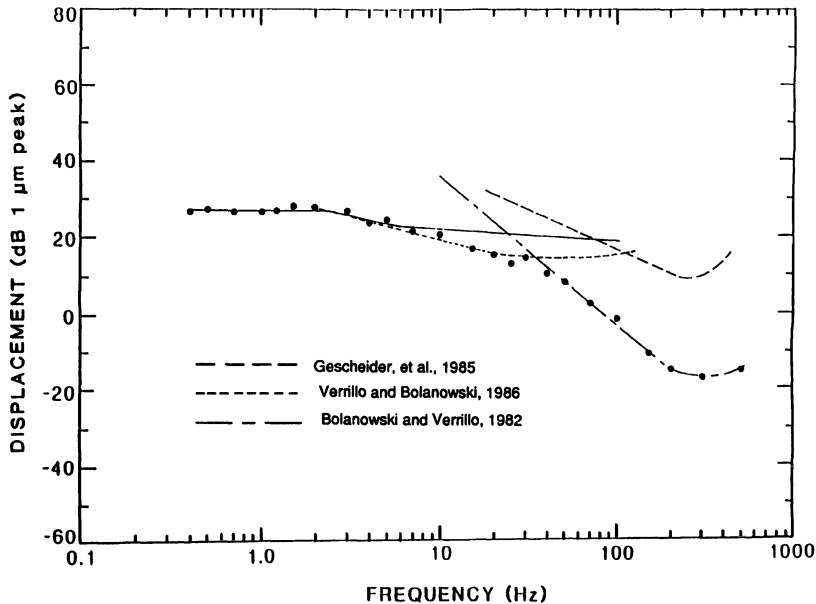


Fig. 6 The averaged overall threshold-frequency characteristic (•) obtained from 5 observers in response to vibratory stimuli presented with a 2.8 cm^2 contactor to the thenar eminence of the hand. The curves comprise the four-channel model for taction, each being the threshold-frequency characteristics of the various channels: — — —, PC; - - - - -, NP I; — — —, NP II and — — —, NP III. See text for discussion regarding origin of the data. From Bolanowski, et al., *J. Acoust. Soc. Amer.* 84, 1680- 1694 (1988).

Verrillo (1962, 1968) and his colleagues (Capraro, et al., 1979; Gescheider, 1976 and Verrillo and Bolanowski, 1986) have shown previously that thresholds between 10 and 100 Hz are mediated by a channel different than the P channel. It is called NP I and is believed to be mediated by RA fibers (Lund, 1966; Talbot et al., 1968) which purportedly innervate Meissner corpuscles. Through several experimental series (Verrillo, 1962, 1968), NP I, unlike the P channel, has been shown not to possess temporal or spatial summation. The short-dash line in Fig. 6 represents the NP I channel based on current knowledge.

The presence of the breakpoint between the low- and middle-frequency portions of the overall threshold function of Fig. 6 suggested the presence of a third channel operating in the lowest-frequency region (0.4-3.0 Hz). A series of experiments was designed to explore this possibility (Bolanowski, et al., 1988). Previous experiments performed in our laboratory showed that at threshold levels of stimulation the channels are independent and do not interact

with each other. For instance, we have shown that an adapting stimulus, which activates only one channel, will have no effect on the threshold response of the other channels (Verrillo and Gescheider, 1977; Gescheider, Frisina and Verrillo, 1979). Using a signal-masking paradigm we also demonstrated channel independence; that is masking stimuli that were within the operating frequency range of one system had no effect on the detection threshold of the other systems (Gescheider, Verrillo, and Van Doren, 1982; Hamer, 1979; Hamer, Verrillo, and Zwislöcki 1983; Verrillo, Gescheider, Calman, and Van Doren 1983; Gescheider, O'Malley, and Verrillo 1983; Gescheider, Sklar, Van Doren, and Verrillo 1985). In general, the channel having the lowest threshold at any given frequency will dominate the detection-threshold response at that frequency. This rule is very orderly

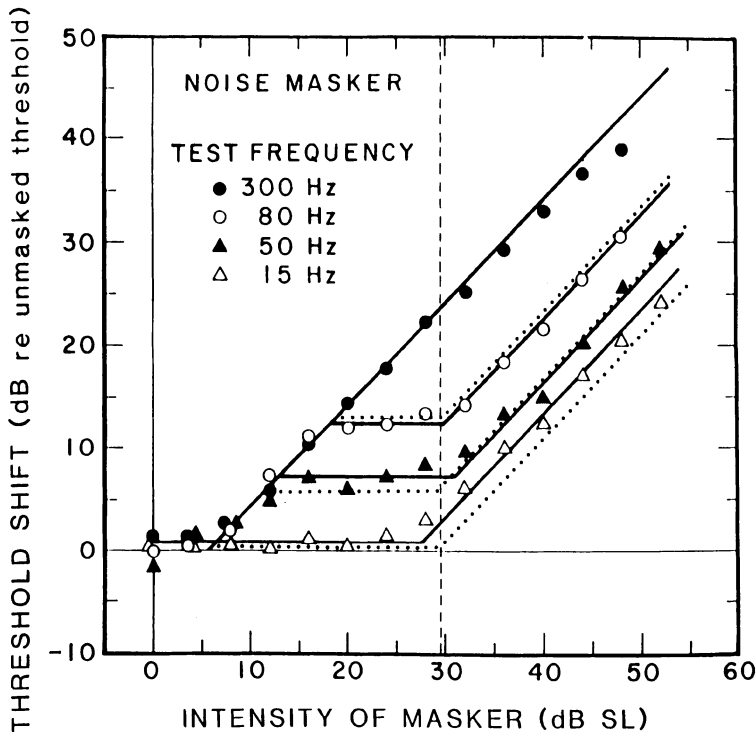


Fig. 7 Masking functions. Vibrotactile threshold shifts as a function of masker intensity. The test frequency was 15, 50, 80, or 300 Hz. The solid lines are linear-regression fits to averaged data from four subjects. The dotted lines are predicted from threshold data. From Gescheider, et al., *J. Acoust. Soc. Amer.* 72, 1421-1426 (1982).

and as a consequence, masking functions, as shown in Figure 7, covering a wide range of intensities and frequencies can be used to predict and uncover frequency responses of

predicted or suspected channels. Although it is not possible to describe how the predictions were made (see Gescheider, Verrillo, and Van Doren, 1982) the figure shows that when masker (1/3 octave noise centered at 250 Hz) and signal (100 Hz, filled circles), are in the same channel, in this case the P channel, the masker produces a loss of sensitivity (masking) in the response to the signal. When the detection signal is set at lower frequencies (80, 50, 15 Hz), optimally activating the NP I system, the signal and masker are in different channels and the masking effect disappears. This is evidenced by the plateau regions of the masking functions. The position of the plateau depends upon the frequency of the signal and the threshold of detection at that frequency. Further increases of intensity ensure that all of the energy is effectively concentrated in the optimal operating range of the other channel. The result is a resumption or continuation of the masking effect, but, now the masking function again describes the effect completely within a single system. The breakpoints in the masking functions are significant in that they define the position at which threshold detection is switched from one channel to another.

By using the masking paradigm we were able to show that the low- and middle-frequency portions of the overall threshold frequency response shown in Fig. 6 were mediated by separate channels. This automatically established the frequency response of the low-frequency channel. We termed this channel NP III and its position and frequency response is shown by the solid line in the figure. Several theoretical arguments explained in detail in Bolanowski, et al., 1988 led us to believe that NP III is mediated by SA I fibers which presumably innervate Merkel-cell neurite complexes.

A third non-Pacinian channel, NP II, the existence of which was originally proposed by Capraro, et al., (1979) has been shown by Gescheider, et al. (1985) to operate in the vibratory-frequency range similar to the P channel but at a much lower sensitivity. In these studies, the channel was defined by desensitizing the P channel through the use of a small stimulus area (i.e., minimizing spatial summation) and by using a masking paradigm to deactivate NP I. The position and response profile of NP II is given by the long-dashed line in Fig. 6. Since only four fiber types have been found in glabrous skin, we concluded by elimination that NP II is mediated by SA II fibers which connect to Ruffini end organs.

In addition to manipulating the mechanical aspects of the stimulus, we varied the temperature of the skin systematically in order to determine if temperature had any effect on the response characteristics of the four channels (Bolanowski and Verrillo, 1982; Verrillo and Bolanowski, 1986). We found that the threshold response of the P channel was greatly affected by temperature, both in its sensitivity level and its frequency of maximal sensitivity

which shifted upward with increasing temperature. The effect of temperature on the NP II channel was also clear-cut but less than that of the Pacinian channel on both hairy and glabrous skin. In general, this population of receptors loses sensitivity as the temperature decreases. The effects of temperature on the other psychophysical channels (NP I and III) were not as clear-cut. The results and subsequent statistical analysis of the results showed that the NP I and NP III channels were clearly affected by temperature changes, but it was not possible to determine the exact manner of the effects because of complex interactions between temperature and stimulus frequency.

A direct comparison of psychophysically-determined average threshold measurements and thresholds obtained in physiological experiments is shown in Fig. 8. The four panels represent the four receptor types and the criteria that meet the needs of the four psychophysical channels described below. The psychophysical data are replotted from Fig. 6. The psychophysical channels shown in the four panels are: A, Non-Pacinian I (NP I); B, Pacinian (P); C, Non-Pacinian III (NP III); and D, Non-Pacinian II (NP II). The physiological data were obtained via the analysis of the data of Johansson, et al. (1982) as detailed above. The solid points and lines represent the physiological results of the four fiber types as shown in the four panels of the figure, namely: A, RA; B, PC; C, SA-I; and D, SA-II fibers. The response criteria selected for the different fiber types were: A, 1 impulse/stimulus; B, 4 impulses/stimulus; C, 0.8 impulse/sec and D, 5 spikes/sec (see Bolanowski, et al., 1988 for an in-depth discussion regarding why these particular criteria were selected). The additional curve (solid line without data points) in panel B is the average results obtained from 5 Pacinian corpuscles isolated from cat mesentery and kept at 33°. The correlations between the physiological and psychophysical results in most cases is good. The matched results are not perfect, but they are close enough to suggest that each psychophysical channel has its own neural substrate and that all four fiber types can contribute to tactile sensation. Several factors can explain the differences between the psychophysical and physiological results including temperature effects, the presence of spatial and temporal summation for certain channels, and the physiological variability in responses. A more thorough discussion regarding these aspects can be found in Bolanowski, et al. (1988).

As we have mentioned before, in every experiment we have ever performed, we have never been able to demonstrate interaction among the P and NP channels at threshold levels of stimulation. At threshold, we are convinced that the channels are completely independent, and that the shape of the overall psychophysical characteristic is determined by the channel most sensitive at any given frequency. At suprathreshold levels of stimulation, however, the picture is certainly more complex. In some psychophysical tasks, independence of channels

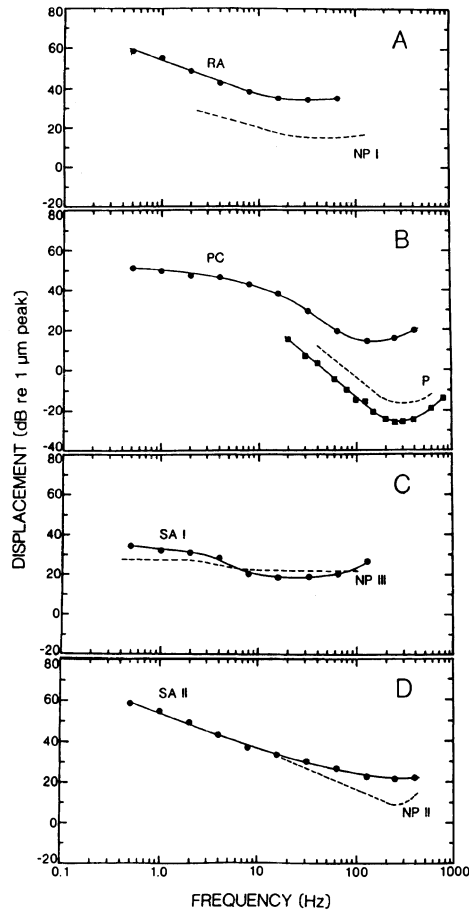


Fig. 8 Relationship between physiologically measured frequency characteristics of different fiber types (A, RA; B, PC; C, SA I and D, SA II) and psychophysically obtained threshold-frequency characteristics (A, NP I; B, P; C, NP III and D, NP II). Neurophysiological data points are interpolations and extrapolations of the average results presented by Johansson, et al., (1982) for selected response criteria: A, 1 impulse/stimulus; B, 4 impulses/stimulus; C, 0.8 impulses/sec and D, 5 spikes/sec. For the P channel (B), an additional physiological curve (Pacini corpuscle) has been plotted. This curve is the average response of excised Pacini corpuscles ($N=6$) maintained at 33°C and for a response criterion of 4 impulses occurring during the central 200 msec of a 300 msec stimulus burst. From Bolanowski, et al., *J. Acoust. Soc. Amer.* 84, 1680- 1694 (1988).

is preserved, but in others there is clear evidence of interaction among the channels. The phenomenon of enhancement, in which a conditioning stimulus has the effect of increasing the perceived intensity of a second stimulus, shows very clear evidence of no interaction between channels (Verrillo and Gescheider, 1975; Gescheider, Verrillo, Capraro, and

Hamer, 1977). If the conditioning stimulus and the test stimulus are in the same channel, the effect of enhancement is dramatic. However, if the two stimuli activate different channels, enhancement disappears. On the other hand, if the subject is asked to estimate the summed, overall, subjective intensity of both conditioning and test stimuli, there is clear evidence of interaction between channels. When the conditioning and test stimuli are set at frequencies that optimally excite different channels, the combined effect of subjective intensity is greater than that of either channel alone. The interaction between channels in these experiments is quite clear.

In another set of experiments, we used a method called absolute magnitude estimation (Stevens, 1957; Hellman and Zwislocki, 1961; Zwislocki and Goodman, 1980; Verrillo, Fraioli, and Smith, 1969) in which the subject is asked to assign numbers perceived to match the subjective magnitude of the sensory experience. In our experiments, we asked subjects to estimate the perceived intensity of vibrotactile stimuli at temperatures known to affect the cutaneous mechanoreceptor channels, either by selectively increasing or decreasing activity in them. Our preliminary data show that the growth of perceived intensity of vibration on the skin is affected indeed by the temperature at the surface of the skin. Furthermore, there are features of the growth functions that suggest that as the physical intensity of the stimulus is increased, there is a sequestering of inputs from the less - sensitive channels. Although these experiments are still in progress, the results agree with the model of four independent channels. This suggests that at suprathreshold levels the code for perceptual quality may be considerably more sophisticated and complicated than had been realized previously, requiring that several channels contribute the information available for this purpose. In other words, fundamental qualities of sensation such as "pressure", "flutter", and "vibration" may contribute to form the many sensory attributes ascribed to the somatosensory system including form, texture, and borders. One implication of the four-channel model is that, before a true understanding of the manner in which sensory experiences such as "roughness", "softness", and a myriad of other sensations, can be achieved, it may be necessary to establish, across all receptor types, the psychophysical and physiological criterion responses that signal a sensory event both in single and in multiple channels. The development of multichannel-stimulation techniques may also be necessary in order to engage the requisite channels. This will be especially important for purposes of developing better devices as surrogate inputs for vision and audition, and for designing appropriate sensory-feedback systems for limb prostheses and robots.

References

- Bolanowski, S.J., Jr.: Intensity and frequency characteristics of Pacinian corpuscles. Institute for Sensory Research, Syracuse University, Syracuse, NY. Ph.D. dissertation and Special Rep., ISR-5-20. 1981
- Bolanowski, S.J., Jr., Gescheider, G.A., Verrillo, R.T., and Checkosky, C.M.: Four channels mediate the mechanical aspects of touch. *J. Acoust. Soc. Am.* 84, 1680 - 1694 (1988)
- Bolanowski, S.J., Jr., and Verrillo, R.T.: Temperature and criterion effects in a somatosensory subsystem: A neurophysiological and psychophysical study. *J. Neurophysiol.* 48, 836 - 855 (1982)
- Capraro, A.J., Verrillo, R.T., and Zwislocki, J.J.: Psychophysical evidence for a triple system of mechanoreceptors. *Sens. Processes* 3, 334 - 352 (1979)
- Cauna, N. and Mannan, G.: The structure of human digital Pacinian corpuscles (*Corpuscula Lamellosa*) and its functional significance. *J. Anat.* 92, 1-25 (1958)
- Chouchkov, C.N.: Ultrastructure of Pacinian corpuscles in men and cats. *Z.F. Mikro-anat. Forsch.* 83, 33-46 (1971)
- Gescheider, G.A., Frisina, R.D., and Verrillo, R.T.: Selective adaptation of vibrotactile thresholds. *Sens. Processes* 3, 37 - 48 (1979)
- Gescheider, G.A., O'Malley, M.J., and Verrillo, R.T.: Vibrotactile forward masking: Evidence for channel independence. *J. Acoust. Soc. Am.* 74, 474 - 485 (1983)
- Gescheider, G.A., Sklar, B.F., Van Doren, C.L., and Verrillo, R.T.: Vibrotactile forward masking: Psychophysical evidence for a triplex theory of cutaneous mechanoreception. *J. Acoust. Soc. Am.* 78, 534 - 543 (1985)
- Gescheider, G.A., Verrillo, R.T., Capraro, A.J., and Hamer, R.D.: Enhancement of vibrotactile sensation magnitude and predictions from the duplex model of mechanoreception. *Sens. Processes* 1, 187 - 203 (1977)
- Gescheider, G.A., Verrillo, R.T., and Van Doren, C.L.: Predictions of vibrotactile masking functions. *J. Acoust. Soc. Am.* 72, 1421 - 1426 (1982)
- Hamer, R.D.: Vibrotactile masking: Evidence for a peripheral energy threshold. Institute for Sensory Research, Syracuse University, Syracuse, NY. Ph.D. dissertation and Special Rep. ISR-5-18. 1979
- Hamer, R.D., Verrillo, R.T., and Zwislocki, J.J.: Vibrotactile masking of Pacinian and non-Pacinian channels. *J. Acoust. Soc. Am.* 73, 1293 - 1303 (1983)
- Hellman, R.P., and Zwislocki, J.: Some factors affecting the estimation of loudness. *J. Acoust. Soc. Am.* 33, 687 - 694 (1961)
- Johansson, R.S., Landström, U., and Lundström, R.: Responses of mechanoreceptive afferent units in glabrous skin of the human hand to sinusoidal skin displacements. *Brain Res.* 244, 17 - 25 (1982)
- Poláček, P. and Mazanec, F.: The distribution of myelin on nerve fibers from Pacinian corpuscles. *J. Physiol.* 129, 167-176 (1966)
- Sinclair, D.: Cutaneous sensation. London: Oxford University Press 1967
- Spencer, P.S. and Schaumburg, H.H.: An ultrastructural study of the inner core of the Pacinian corpuscle. *J. Neurocytol.* 2, 217-235 (1973)
- Stevens, S.S.: On the psychophysical law. *Psychol. Rev.* 64, 153 - 181 (1957)
- Vallbo, A.B., and Hagbarth, K.-E.: Activity from skin mechanoreceptors recorded percutaneously in awake human subjects. *Exp. Neurol.* 21, 270 - 289 (1968)
- Van Doren, C.L.: A model of spatiotemporal tactile sensitivity linking psychophysics to tissue mechanics. *J. Acoust. Soc. Am.* 85, 2065 - 2080 (1989)
- Verrillo, R.T.: Investigation of some parameters of the cutaneous threshold for vibration. *J. Acoust. Soc. Am.* 34, 1768 - 1773 (1962)
- Verrillo, R.T.: Effect of contractor area on the vibrotactile threshold. *J. Acoust. Soc. Am.* 35, 1962 - 1966 (1963)
- Verrillo, R.T.: Temporal summation in vibrotactile sensitivity. *J. Acoust. Soc. Am.* 37, 843 - 846 (1965)

- Verrillo, R.T.: Vibrotactile sensitivity and the frequency response of the Pacinian corpuscle. *Psychon. Sci.* 4, 135 - 136 (1966)
- Verrillo, R.T.: A duplex mechanism of mechanoreception. In: *The Skin Senses*. (D.R. Kenshalo Ed.) Springfield, IL: Thomas 139 - 159. 1968
- Verrillo, R.T., and Bolanowski, S.J., Jr.: The effects of skin temperature on the psychophysical responses to vibration on glabrous skin and hairy skin. *J. Acoust. Soc. Am.* 80, 528 - 532 (1986)
- Verrillo, R.T., Fraioli, A.J., and Smith, R.L.: Sensation magnitude of vibrotactile stimuli. *Percept. Psychophys.* 6, 366 - 372 (1969)
- Verrillo, R.T. and Gescheider, G.A.: Enhancement and summation in the perception of two successive vibrotactile stimuli. *Percept. Psychophys.* 18, 128 - 136 (1975)
- Verrillo, R.T. and Gescheider, G.A.: Effects of prior stimulation on vibrotactile thresholds. *Sen. Processes* 3, 37 - 48 (1979)
- Verrillo, R.T., Gescheider, G.A., Calman, B.G., and Van Doren, C.L.: Vibrotactile masking: effects of one - and two-site stimulation. *Percept. Psychophys.* 33, 379 - 387 (1983)
- Zwislocki, J.J., and Goodman, D.A.: Absolute scaling of sensory magnitudes - a validation. *Percept. Psychophys.* 28, 28 - 38 (1980)
- Zwislocki, J.J., Maire, F., Feldman, A.S., and Rubin, H.J.: On the effect of practice and motivation on the threshold of audibility. *J. Acoust. Soc. Am.* 30, 254 - 262 (1958)

Acknowledgements

This research was supported in part by Grants NS 23933, R01 DC-00098, and P01 DC-00380, National Institutes of Health, U.S. Department of Health and Human Services.

Borrowing Some Ideas From Biological Manipulators to Design an Artificial One

Vincent Hayward

McGill Research Center for Intelligent Machines
McGill University, 3480 University Street
Montréal, Québec Canada H3A 2A7

Abstract. The design of robotic manipulators is a difficult question because most of the traditional disciplines needed for the design of robots, like kinematics and dynamics, are mostly analytic and have little synthetic power. We first discuss design seen as a generative process and suggest that analogy is a powerful design method. Then a spherical mechanism actuated in parallel with a large workspace that can be used to construct a complete limb is discussed. The design synthesis is performed by translating ideas borrowed from the design of biological manipulators.

1 Introduction

Commercially available robot manipulators exhibit a degree of elegance and adequacy which is far from approaching what can be observed in biological manipulators. Hence, seeking inspiration from Nature remains quite an appealing approach. In fact, even the most application oriented industrial manipulators always bear some degree of resemblance with human arms: a sequence of articulated bodies with a distinguishable shoulder, elbow and wrist, see Figure 1 for example; while submarine manipulators, for another example see Figure 2, recall crustacean limbs.

This suggests that despite the claim that artificial manipulators really must match their applications and that no valid reason exists for using anthropomorphism (and zoomorphism), the models of Nature remain, consciously or not, an inexhaustible source of inspiration.¹

Design of manipulators entails a decision making process which concerns many attributes of the device, encompassing materials, assembly methods, mechanical structures, computational structures, sensor, motor and motion transmission technologies, and so-on, to achieve a desired level of functional capacity. The organic quality of biological systems, which any person engaged in engineering research can easily appreciate, is far from being achieved by any technological systems, except perhaps by those artifacts which have been developed and refined over centuries. Such examples can be found in hand tools and musical instruments. The violin, for instance, achieves the integration of

¹As J. Phillips puts it: "There is of course no reason to believe that robots (which are machines) should resemble us or animals, both of which are also machines; but the occurrence of anthropomorphism in our thinking and the consequent discussion about its appropriateness in design is almost inescapable" [14].

several of the above mentioned aspects of design at an extraordinary level of harmony.

The general objective of robot manipulator design is to devise a machine capable of (1) displacing tools within the largest possible amount of space while minimizing spatial intrusion or interference with the environment, and (2) imparting forces and torques onto the environment in a delicate and controlled fashion once a desired collision occurs, while (3) at the same time it is also capable of moving in free space at high velocity [4]. The problem stated above separates into two parts. The givens which are decided by the *design* and the *controls* which confer properties not exhibited by the original device.

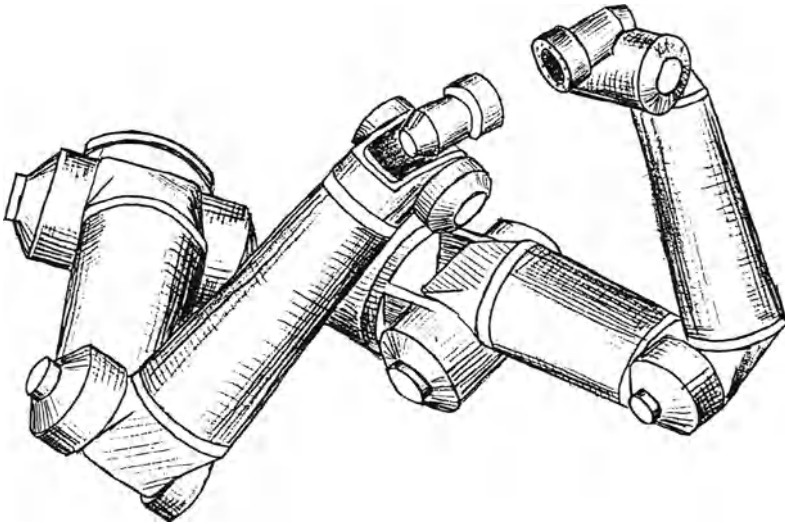


Fig. 1. Pair of manipulators designed by Robotics Research Inc.

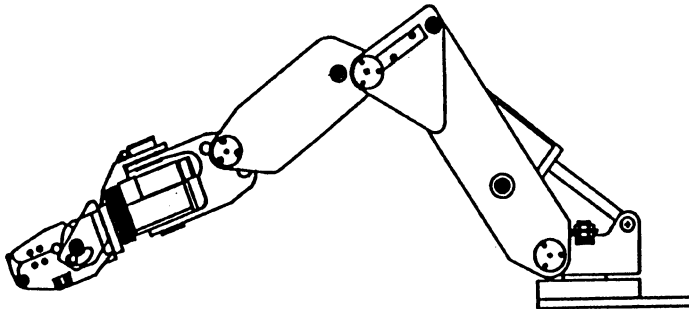


Fig. 2. Sketch of a submarine manipulator built by International Submarine Engineering Ltd.

Clearly, the properties defined by design set bounds on what can be achieved by control. In the sole domain of kinematics, it is not the goal of robotics research to find all possible arrangements (which may be a the goal of the Theory of Mechanisms), but to find the most relevant ones for manipulation.

The largest amount of effort in robotics research has been concerned with the development of analytical tools such as kinematics and dynamics, disciplines that rest on well established physical principles. However, work on design still relies mostly on intuition because the synthetic power of these disciplines is difficult to exploit.

The design of biological systems transcends human comprehension and is expected to remain as such well beyond the foreseeable future. It is however clear that the observation of salient features of examples found in Nature can lead to insights readily usable in technological systems. This paper attempts to suggest that Nature's example can point to kinematic and structural suggestions quite applicable to current technology and which are directly derived from anatomical features observed in natural limbs.

Contemporary and historical examples of this abound. Robotics takes its roots in the development of machines to extend human capacities. Thus, the history of robotics may be traced back at least to the Bronze age with the discovery of levers and wheels (rotary motion). Through-out the ages, developments have been contributed by various civilizations. Examples come from the Sumerians, Greeks, Romans, the Renaissance, the Age of Enlightenment, the Industrial Revolution, and not even including the less known in the West Asiatic Cultures, in a pattern chronologically aligned with the history of technology. In the honor of the province of Tuscany which hosted this meeting, Leonardo da Vinci should be singled out as an illustrious precursor of the design methodology based on the observation of Nature. The following example is particularly relevant to the theme of this paper.

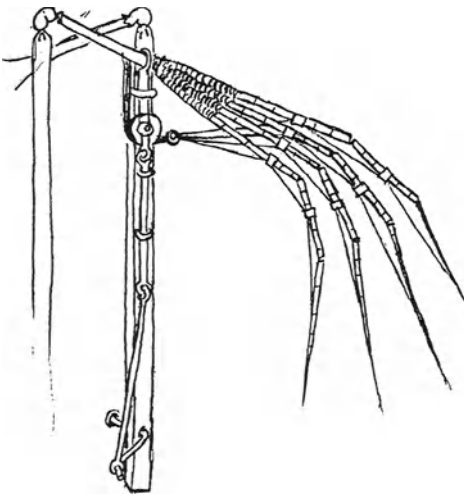


Fig. 3. This study suggests the emulation of bird wings [Cod. Atl. f. 308r.-a].

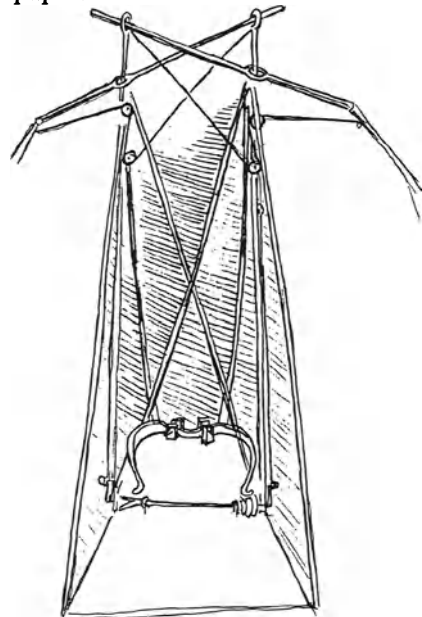


Fig. 4. Leonardo envisaged springs to store energy in this "Ornitottero" [Cod. Atl. f. 314r.-b.].

Leonardo made extensive studies about bird wings in an attempt to emulate flight, see Figure 3, for example. As far as we know, these attempts were unsuccessful. He probably convinced himself that flight emulation could not be achieved by wing flapping mechanisms actuated by human muscular power and imagined to use springs to provide power, see Figure 4. It is nevertheless likely that the attempt to utilize aerodynamic forces in a more efficient manner led him to imagine the famous “air screw”, see figure 5 [3].

Since Nature optimizes her designs for reasons which we are not fully aware of, there is limited justification for attempting faithful emulation of these designs. Rather, the approach might be the re-exploitation of certain design features found in Nature. It is manifest that biological manipulators are not optimized for many tasks of interest: a human arm is obviously ill suited to intervene in a nuclear reactor core. This does not mean that structures observed in Nature cannot be re-utilized.

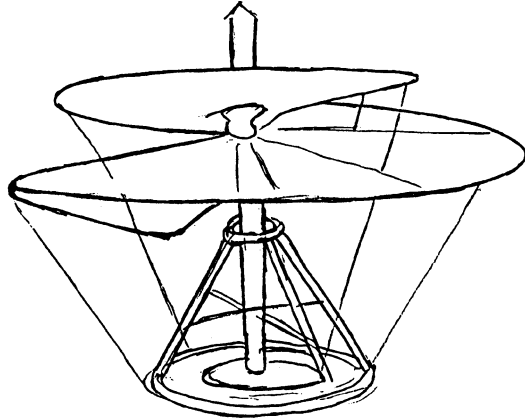


Fig. 5. Leonardo's "Air Screw" [Ms.B.f. 83v.]

An exploratory study of redundancy was our motivating factor for the arm design described later in this paper. It was recognized that redundancy is not only desirable, but necessary to the design of general purpose manipulators [6].

2 Design as Problem Solving

Ex nihilo nihil fit, design ideas can most of the time be traced back to some earlier attempts.² In general, design, seen as a problem solving activity, is very unconstrained. It has been observed that it can be described more like a process-driven activity rather than an optimizing activity. According to Simon, the design 'process' is picked by the designer according to a complex set of reasons while the goal may remain fuzzy [18].

Design proceeds by generation alternative designs are produced in large numbers until one of them satisfies a set of criteria. Only then, can an analytical optimizing activity take place. In the case of manipulators, only a surprisingly small number of design processes have been utilized by the industry, the result being a limited number of design styles, possibly because the robot manipulator technology is quite recent. It is interesting to look back for a moment at the past few decades during which industrial manipulators were developed. Apart from a few notable exceptions, current design concepts more

²For R. Buckminster Fuller: "When you and I speak of design, we spontaneously think of an intellectual conceptualizing event in which the intellect first sorts out a plurality of elements and then interarrange them in a preferred manner." [1].

or less follow the machine tool engineering tradition. This can be observed for robots used in the automotive industry.

Most of those manipulators are designed for high positional accuracy and high rigidity, which makes them adequate for machine-tool-like applications. A number of difficulties are created when it is attempted to use these devices for other kinds of tasks, particularly those involving the control of forces when in contact with the environment.

Among all existing kinematic structures, a four-bar mechanism for inner joints augmented by a three axis wrist with intersecting axes has emerged over time as the vastly dominant structure, as in a kind of a Darwinian evolution process. Similarly, one other kinematic structure known as the SCARA design (Selective Compliance Assembly Robot Arm) is overwhelmingly used in precision assembly applications because of its adequacy for the task (dynamic and kinematic decoupling along the vertical and horizontal directions).

Manipulator design occurs trying to satisfy an open set of constraints resulting in part from the laws of Nature, some of which are captured by the equations of kinematics and dynamics. Kinematics and dynamics have little synthetic power: they permit a designer to improve a proposed design through analysis or optimization, or to determine local features such the shape of cams. Sometimes, qualitative exploration of many arrangements in order to reach a functional goal is possible as demonstrated by Salisbury in the context of arm manipulation [16]. Other constraints result from technological feasibility. These are of course difficult to obtain since they depend on the accuracy of available information, the risk involved in creating new technologies, and the rate of improvement. The remainder of the constraints encompasses a set of desired properties which can be quite arbitrary. These are decided upon by the designer for reasons that may have to do with experience, tradition, personality, wit, corporate image, budget, trends, fashion, and so-on.

Vastly different motivations may be noticed in discussions pertaining to robotic designs, and once again two views can be opposed. The analytical, proof by existence, approach: "Nature produces systems which utilize real hardware that operates according to physical principles...the intent [of the design] is not to imply that the development of such systems will be an easy task, only that such systems can be developed" [11]; and the synthetical, task oriented, approach: "we feel that what is needed is a *medium-complexity* end effector: a device that combines the ease of control characteristic of the simple grippers with some of the versatility of the complex hands" [20].

As a result, an all-encompassing design goal can never be formalized; instead, as commented above, a generative method is selected. Possibilities are matched against the criteria that have been decided upon in advance. Unpromising alternatives of the successive versions are filtered in a process which is reminiscent of a technique known in artificial intelligence as "means-end analysis." In this technique, not only immediate choices are made to progress toward a goal, but also choices about the operators that are likely to lead to progression. The definition of quantitative criteria may help to automate part of the search process. The final goal is known once successive generations have

filtered through the constraints. However, it is unlikely that this design process will ever be reduced solely to an explicit search process, or to an optimizing process, game theoretical or otherwise.

Optimality is difficult to include in the robot design activity, because optimality entails the existence of a well defined objective function, which opposes the requirement to create a general purpose machine. It is impossible to think of such a function since the space over which this function would be defined cannot be known before the end-result of the design process has been satisfactorily described. Nonetheless, a design can be declared optimal with respect to a particular mathematical model and a particular criterion defined over the variables of this model. The relevance of the model is then of course an essential question. It has been our experience that oversimplification leads to physically non realizable structures [12].

A common methodology first entails the creation of generic modules which can be instantiated into a collection of devices having scaled properties (size, power and so on). The advantages of such an approach are well known and discussed at length in the computer science literature. The principles put forward in computer science are standardization (interface rules), polymorphism (hidding implementation), and composition (larger blocks made of smaller ones). They promote abstractions, reliability, ease of maintenance, and top-down design. Clearly, these principles significantly apply as well to electro-mechanical design. The second part of this methodology requires a decision upon a framework structure describing how modules inter-relate. In dealing with complexity, hierarchical organizations are often proposed.

3 Overall Approach

Some of the properties observed in biological manipulators that can be put to use in technological designs are now discussed. The most general observations fall in two categories: (1) on actuation and (2) on kinematics and structures. It is the purpose of this study to explore the second category in greater details.

Limbs in Nature come in two varieties: endo-skeletons and exo-skeletons. In the endo-skeleton case, most of the material used passively (bones) is located *inside* the material used actively (muscles), whereas the opposite situation is observed in the exo-skeleton case (shells). This opposition is also observed to some degree of approximation in the distribution of material used in compression is compared to that of material used in extension.

So far, the design of artificial manipulators has followed mostly the exo-skeleton case. In contrast, we will follow here the endo-skeleton path (vertebrae) simply following the intuition that natural endo-skeletons seem more agile than the exo-skeleton ones (crustaceans).

The most identifiable anatomical elements (anatomy deals with structure and morphology) are at a macroscopic scale, in the endo-skeleton case: muscles, tendons, ligaments, bones, and synovial joints. These elements correspond to a separation of mechanical and structural functions: extension, compression, mobility. We will also attempt of incorporate this separation in our design.

A great deal of mobility in biological endo-skeletons limbs is achieved

through joints which approximate revolute (elbow, knee) pairs or spherical pairs (e.g. shoulder, hip, eye). These correspond to two symmetries that allow continuous surface contact under motion: axial symmetry (revolute) and point symmetry (sphere). The other pairs (planar, prismatic and screw) are not found in natural limbs. An essential element of biological limbs is the spherical pair. Biological systems actuate spherical pairs using parallel actuation. The technological analogy is the parallel manipulator discussed below in greater details.

The traditional design of manipulators is based on a completely serial design: a succession of links and joints. Serial manipulators lead to accumulation of errors, lack of rigidity, low natural frequency that can be counteracted with parallel designs [10]. Despite the drawbacks of such an approach, it is the most commonly found structure. One of the reasons might be that their models lend themselves to easier analytical studies than those of parallel manipulators.

The serial robot manipulator technology mostly uses massive metallic structures designed to counteract the cantilever effect. An direct consequence is a resulting very poor weight/load ratios due to the "pyramidal effect": Proximal joints must be designed to drive and support the sum of the distal links and joints.

The principal advantage of serial manipulators is the amount of workspace and the minimization of spatial intrusion. Clearly, what is needed is a combination of serial and parallel kinematics. It is not surprising that natural limbs are partly serial and partly parallel: the skeleton-muscle system creates many closed kinematic loops (quite complex to analyze), yet there is an amount of seriality to yield workspace (arm-forearm-hand).

A complicated problem in the design of manipulators is the integration of actuators and sensors into the overall structure. Nature integrates sensors directly within the actuators at the microscopic scale and provides motion transmission devices with very small losses (tendons and sheaths). Of course, this idea has been utilized in the design of manipulators and mechanical hands despite numerous practical difficulties. A parallel kinematic structure with linear actuators can be viewed as a deformable truss.

In such a truss design, actuators and sensors can be made parts of the structure, thus achieving a high degree of integration that characterizes biological designs. Yet, the various parts of the structure can be made easily accessible and similar to others. This promotes modularity and interchangeability [8].

An additional remark adds weight in favor of the endo-skeleton case. Regardless of the structure which is chosen, position, velocities and forces need to be measured for control of manipulators. It is a fact of mechanics that the greatest amounts of velocity and smallest amount of forces in a manipulator in action will manifest themselves at the exterior parts of the structures. This suggests that force production elements as well as sensors should be placed as close to the external regions as possible. Thus passive elements should be placed inside to complete the structure which is made possible by the use of trussed structures.

Truss structures have also interesting properties which are quite appealing

for limbs designs: the load on parts of the structure and on joints is always axial, they can be made out of a small vocabulary of elements, and a great deal is available on the design of such structures.

4 Topological and Geometrical Observations

Mechanisms may become “singular”. In fact, the map from input coordinates (joint variables) to output coordinates (active link coordinates) displays singularities. To better illustrate that concept we will use topological terms as proposed by Burdick [2]. Homotopy allows to view mechanism at “order zero”, to describe qualitatively their kinematic properties. This can be easily grasped by considering a two link manipulator, Figure 6.

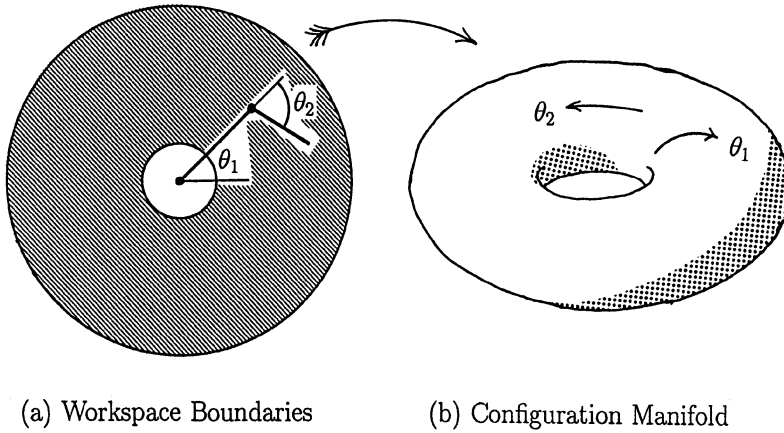


Fig. 6. Two link manipulator (a) and its configuration manifold (b), created by “stitching” two sheets together: $0 < \theta_2 < \pi$ and $-\pi < \theta_2 < 0$.

Singularities, described as critical points of the configuration manifold, come in two types. Separating singularities divide the configuration manifold into sheets such that any motion from one to the other must traverse a locus of singularities. Non-separating singularities simply create “holes”. These singularities are situated inside the workspace and motion involving constraints placed in the end-effector motions must avoid the surrounding region.

The workspace of robot mechanisms is determined by three factors: self-interference of parts, travel limits of actuators, and one special locus of singularity of the separating type. In the case of a planar two links manipulator, it is easy to see that this locus is a circle centered at the first joint. There is also a geometric interpretation of singularities. In the case of serial manipulators, singularities occur, for example, when the axes of revolute joints align because two joints become mutually redundant. The manipulator becomes “locked” for motions around a direction perpendicular to the mutual axis due to loss of a degree of freedom.

As an example, we will illustrate this interpretation on the advanced manipulator designed by Salisbury and Townsend described in this proceedings. This arm, the geometry of which is seen Figure 7, has of two elongated links.

It has been designed so as to be able to utilize the entire surface of its links in contact tasks. Thus, a complete mobility of both links is essential. From a geometrical view-point, the objective of orienting arbitrarily the two links in space is completely achieved (four parameters, four joints). However, the existence of a "hole creating" singularity when joint 1 and 3 are aligned prevents full usage of the arm within its workspace, although it can freely maneuver around it.

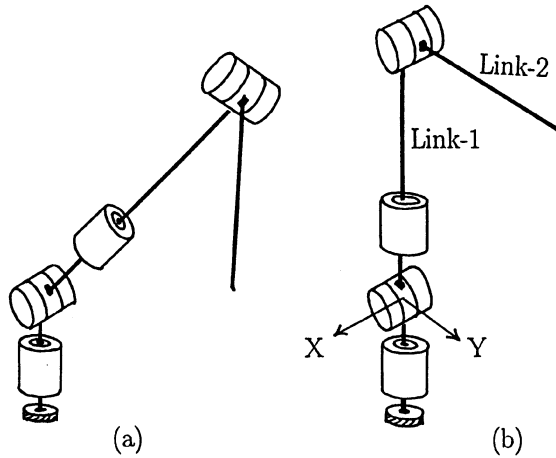


Fig. 7. (a) Geometry of WAMS.
(b) Link-1 cannot rotate around axis Y.

The problem of loss of mobility of serial manipulators can be treated with supplementary joints which enhance the global mobility of the mechanism in such a way that local loss of mobility can be counteracted with its kinematic redundancy [6]. The example of a four revolute joint mechanism which provides full orientation capability has been worked out by Long and Paul [13]. This strategy has only limited applicability for a number of reasons. Adding more serial joints only increase the problems that affect serial manipulators such as accumulation of errors, and degradation of dynamics that have been alluded to earlier.

In addition, augmenting the number of revolute joints does not remove any singularities for reasons that are clear from Burdick's topological arguments. In fact, the more serial joints are added, the more complex the topological map of the manipulator becomes and the more complex the control and programming become. Thus this possibility for designing a highly dextrous manipulator has been discarded. We now turn our attention to parallel manipulators, since it is the intention to include them in the design.

As described by Hunt [10], for parallel manipulators, singularities also occur in special geometric situations such that motions cannot be controlled by the actuators (e.g. piston and crank system when the crank is fully extended or retracted). In other terms, the actuated joint velocities vanish for finite motions of the mechanism.

It is possible to classify the singularities of parallel manipulators into three types [5]: the singularities of the sheet separating type when one of the serial subchain of the mechanism is singular—loss of mobility—; the singularities of Hunt type—loss of controllability—see Figure 8, or both. The third case occurs only for special configurations which cause two singularities to meet,

which can be avoided by design. The important observation that we will use in the next section is that the loss of controllability for a parallel mechanism occurs in general inside its workspace, but may retain mobility in large portions of workspace. Of course, biological manipulators do not escape this laws.

The human shoulder for instance has a very large workspace. Its large controllability region can be attributed to redundancy in actuation and this will lead us to utilize a similar method to eliminate singularities of Hunt type in large regions of a parallel manipulator. The results of the above discussion are now utilized to formulate the design of a mechanism that does not display singularities in large portion of its workspace.

5 Kinematic Synthesis

Consistent with the goal to achieve a large workspace and limited spatial intrusion, it seems difficult to avoid the general architecture which consists of two elongated links assembled by a revolute joint. Such a manipulator, using a three revolute joints assembly at each end, was first described in the 70's by Takase, Inoue and Sato [19] and later discussed by Hollerbach [9]. As shown by Yoshikawa [22], its kinematic decoupling simplifies enormously many aspects of the control, in particular when the task prescribes the hand motions while collisions need to be avoided. Nevertheless, as commented before, this architecture still possesses "hole creating" singularities which defeat some of its advantages.

In addition, such a manipulator requires to cascade seven joints which makes it difficult to obtain good dynamics and accuracy. Following Nature's example, it seems possible to achieve a similar amount of workspace, but using at each end two parallel type mechanisms. This leads to the general architecture on Figure 9.

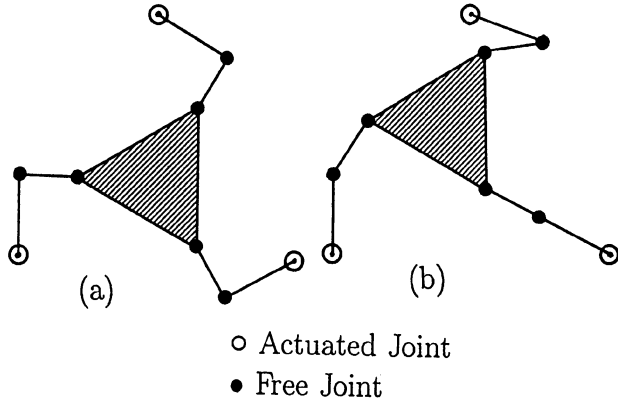


Fig. 8. (a) Loss of controllability. The platform can undergo small rotations while the actuators' velocities vanish. (b) Loss of mobility. The platform is only able to rotate.

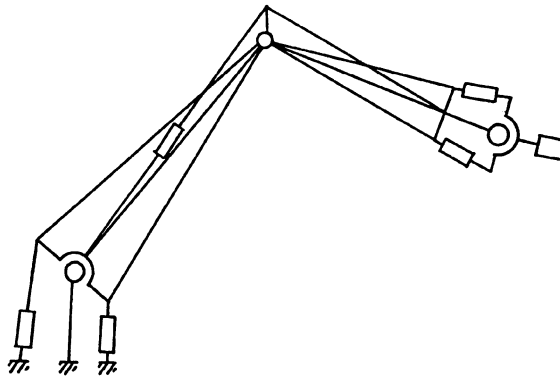


Fig. 9. General architecture.

Such an architecture can only be useful if a sufficient amount of workspace can be obtained from these parallel joints. We have seen that parallel mechanisms made it difficult to achieve good controllable workspace. The main point is that a major source of workspace limitation in parallel mechanisms is due to Hunt type singularities. In fact, this difficulty can be overcome using once again inspiration from biological joints.

For example, the shoulder joint has a large number of muscles to control it. In certain positions, it is clear that some of these muscles cannot contribute to certain motions, but the overall joint is assembled in such a way that when some muscles lose their influence on the output, there are always others to supplement them.

This idea can be readily used in parallel mechanisms. If we look at a simple arrangement of a spherical mechanism, Figure 10, it displays a debilitating singularity right in the middle of its workspace. Because of the underlying topological properties of its kinematic map, this does not depend on the geometry of the mechanism. Regardless of the placement of the actuators, it will always exist. Now consider again a planar type parallel manipulator as shown on Figure 11.

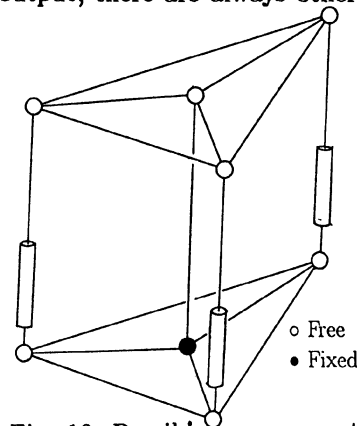


Fig. 10. Possible arrangement.

In the middle of its workspace, the addition of one actuator supplements the loss of controllability. In fact, we have shown that the addition of only one actuator can remove Hunt singularities from a very large portion of the work space from our initial design. The mathematical details of the proof are beyond the scope of this paper, but can be found in [7]. The arrangement shown on Figure 12 possesses a useful range of motion with no self-interference of parts and high and smooth dexterity in the range: $120^\circ \times 180^\circ \times 270^\circ$.

Once physical considerations such as the size and stroke of actuators are taken into account these figures may reduce somewhat. Nonetheless, we have constructed an hydraulically actuated prototype which exhibits a $100^\circ \times 100^\circ \times 180^\circ$ useful range. If desired, it can even be made isotropic, that is optimally dextrous, for several configurations.

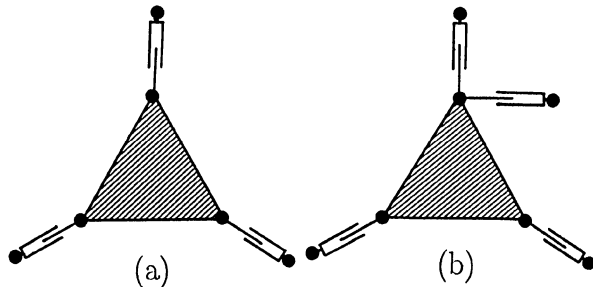


Fig. 11. (a) Hunt type singularity. The reader might agree that it is hard to resist the idea of adding one actuator as in (b).

The measure of dexterity is based on the condition number of the Jacobian matrix of the kinematic map [15]. It has several physical interpretations including mechanism accuracy and a measure of quality for the transmission of forces and velocities from actuators coordinated to output coordinates. Details about kinematic optimization can be found in [12].

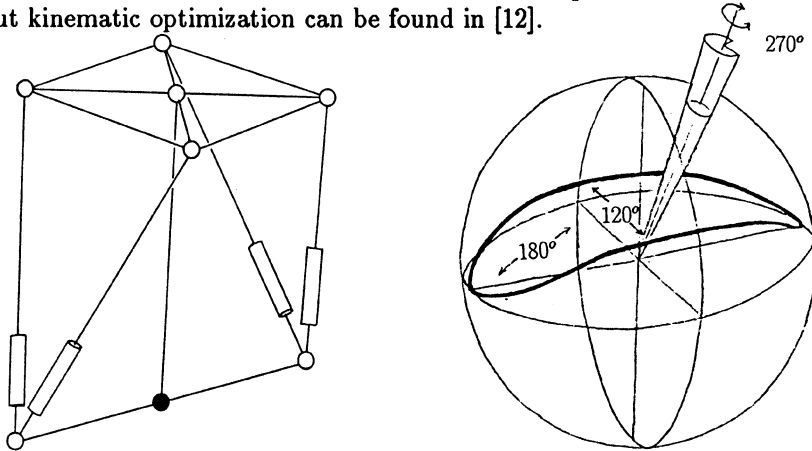


Fig. 12. General concept of the actuator redundant wrist and illustration of its dextrous workspace.

6 A Complete Arm

The integration of the spherical mechanism into a complete arm design will achieve the goal of a creating an arm with limited seriality (three links) and kinematic redundancy as seen from the task (seven freedoms to provide for self-motion that is finite motions with hand fixed).

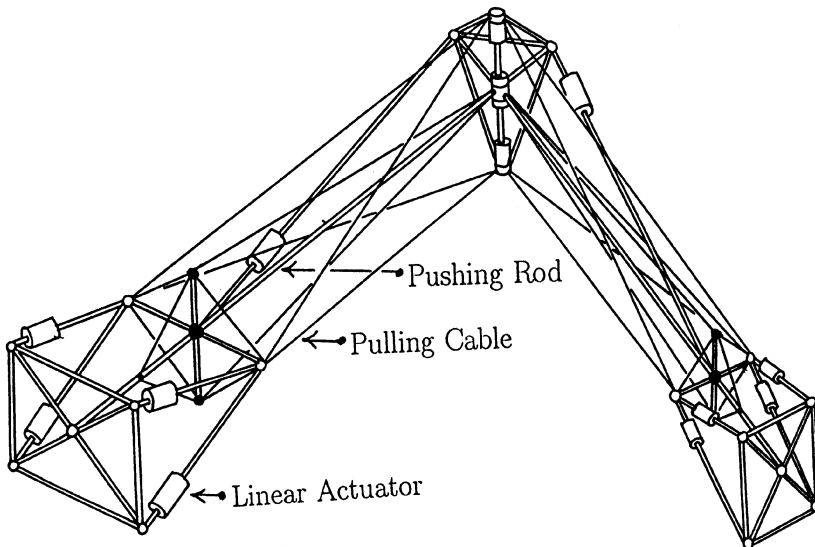


Fig. 13. Concept of complete arm.
This design follows closely Fuller's Tensegrity Principle.

Parallel actuation will lead to high bandwidth and rigidity as well as providing the basis for elaborating a truss assembly. In addition, this manipulator has no "hole creating" singularities since no revolute joints can align, nor Hunt type singularities within a reasonably large workspace. The only singularity left corresponds to the limit of the position workspace when the arm is completely stretched. See Figure 13 for a sketch of the design concept of this arm.

Of course there are as many possible variations around this theme. In particular, it would be particularly interesting to de-locate the actuators of distal links. Some notable successes in this area have already been achieved [17, 21].

7 Conclusion

It has been argued that throughout the history of technology, analogies with biological systems have successfully lead to insights into innovative designs. Many papers in this proceedings will certainly add weight to this idea.

Introspection then has been used to describe a "design process" directed by analogies with biological manipulators aimed at proposing a novel type of robot manipulator which is realizable with existing technology and which possesses a number of desirable properties.

References

- [1] Buckminster Fuller, R. 1985. *Inventions: The patented works of R. Buckminster Fuller*, St. Martin Press.
- [2] Burdick, J. 1989. Kinematic analysis of redundant manipulators: A topological perspective. In *Robots with redundancy: design, sensing and control*, NATO Series, A. Bejczy (Ed.), Springer Verlag, in press.
- [3] M. Chanchi 1984. *Les Machines de Léonard de Vinci*, Becocci Editore, Milano (translated from Italian).
- [4] Goertz, R. C. 1963. Manipulators used for handling radioactive material. *Human Factors in Technology*, Chapter 27, edited by E. M. Bennett, McGraw-Hill.
- [5] Gosselin, C., 1988. Kinematic analysis, optimization and programming of parallel robotic manipulators. *Ph.D. Dissertation*, Dept. of Mech. Eng., McGill University. 1988.
- [6] Hayward V. An analysis of redundant manipulators from several viewpoints. In *Robots with redundancy: design, sensing and control*, NATO Series, A. Bejczy (Ed.), Springer Verlag, in press.
- [7] Hayward, V., Kurtz, R. 1991. Modeling of a parallel wrist mechanism with actuator redundancy. In *Advances in Robot Kinematics*. S. Stifter and J. Lenarcic (Eds). Springer Verlag, 1991.
- [8] Dietrich, J., Hirzinger, G., Gombert, B., and Schott, J. 1989. On a unified concept for a new generation of light-weight robots. In *Experimental Robotics I*, V. Hayward, O. Khatib, (Eds.), Lecture Notes in Control and Information Science 139, Springer Verlag.

- [9] Hollerbach, J. 1985. Optimum kinematic design for a seven degree of freedom manipulator. In *Robotics Research: The Second International Symposium*, H. Hanafusa and H. Inoue (Eds.), 1985, MIT Press.
- [10] Hunt, K. H., 1983. Structural kinematics of in-parallel-actuated robot arms. *ASME, J. of Mechanisms, Transmission, and Automation in Design*, Vol 105, pp. 705-712.
- [11] Jacobsen, S. C., Iversen, E. K., Knutti, D. F., Johnson, R. T., Biggers, K. B. 1986. Design of the UTAH/MIT dextrous hand. *IEEE Conf. Robotics and Automation*.
- [12] Kurtz, R., Hayward, V. 1992. Multi-goal optimization of a parallel mechanism with actuator redundancy. *IEEE Transactions on Robotics and Automation*. Vol. 8, No. 5.
- [13] Long, G. L., Paul, R. P. 1988. Avoiding orientations singularities with a four-revolute-joint spherical wrist. *The Second Workshop on Manipulators, Sensors and Steps Toward Mobility*. Salford, England.
- [14] Phillips, J. 1984. *Freedom in Machinery, Vol. 1: Introducing Screw Theory*, Cambridge University Press, Cambridge.
- [15] Salisbury, J. K., Craig, J. J. 1982. Articulated hands: force control and kinematic issues. *The int. J. of Robotics Research*, Vol. 1, No.1.
- [16] Salisbury, K. 1987. Whole arm manipulation. In *Fourth Int. Symposium on Robotics Research*, R. C. Bolles and B. Roth (Eds.), MIT Press.
- [17] Salisbury, J. K., Townsend, W. T., Eberman, B. S., DiPietro, D., 1988. Preliminary Design of a Whole-Arm Manipulation System (WAMS). *proc 1987 IEEE Int. Conf. Robotics and Automation*, Philadelphia, PA.
- [18] Simon, H. A. 1985. *The sciences of artificial*, MIT Press.
- [19] Takase, K., Inoue, H., and Sato, K. 1974. The design of an articulated manipulator with torque control ability. *Fourth International Symposium on Industrial Robots*.
- [20] Ulrich, N., Paul, R.P., Bajczyk, R. 1988. A medium-complexity compliant end effector. *IEEE Conf. Robotics and Automation*.
- [21] Vertut, J., Marchal, P., Debrie, G., Petit, M., Francois, D., Coiffet, P. 1976. The MA 23 bilateral servomanipulator system. *Proc. of the 24th Conf. on Remote Systems Technology*, Washington. pp. 175-187.
- [22] Yoshikawa, T. 1984. Analysis and Control of Robot Manipulators with Redundancy. *Robotics Research: The First International Symposium*, M. Brady and R. Paul (Eds.), MIT Press.

Mechanical Design for Whole-Arm Manipulation

William T. Townsend and J. Kenneth Salisbury

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA

Abstract

This paper describes the performance requirements and mechanical design of an arm designed and built at MIT for whole-arm manipulation. Whole-arm manipulation began as a research objective to explore the benefits of manipulating objects with all surfaces of a robotic manipulator — not just the fingertips of an attached robotic hand. The need for robust environment contact by all surfaces of the robotic hardware prompted a re-evaluation of traditional manipulator design requirements and spurred the invention of new transmission mechanisms for robots.

1 Introduction – Whole-Arm Manipulation

Salisbury [Salisbury 87, Salisbury 88] introduced the concept of whole-arm manipulation (WAM) to address a broad range of tasks. As a tool for WAM experiments, Townsend [Townsend 88A] designed and built the WAM manipulator, intended for contact and interaction with the environment by using any of its link surfaces. Conventional manipulators, by contrast, must contact the environment with only the inside surfaces of an attached gripper or the fingertips of an attached hand. Often it is useful, if not inevitable, to contact the environment with other parts of the arm.

There are numerous examples where whole-arm manipulation is important. Obstacles that today's robots try to avoid can be used for leverage [West 87] or to guide a robot toward its goal. Furthermore, a human may use his shoulder for mechanical advantage to budge a heavy box, or he may carry firewood between his upper and fore arms by using these limbs as force-controlled grippers. It's hard, for example, to imagine an Olympic-style wrestler winning his match by using only his finger tips! He must control, with tremendous strength, speed, and agility, positions and forces along many parts of his body simultaneously.

Robust, high-performance force control is important to WAM since the system is intended to control contact forces between objects in the environment and any part of its mechanism. This is accomplished by controlling joint torques directly and inferring contact forces rather than measuring them explicitly through a wrist sensor [Salisbury 86]. Figure 1 shows the simplest example of this method of inference. Joint torques τ_1

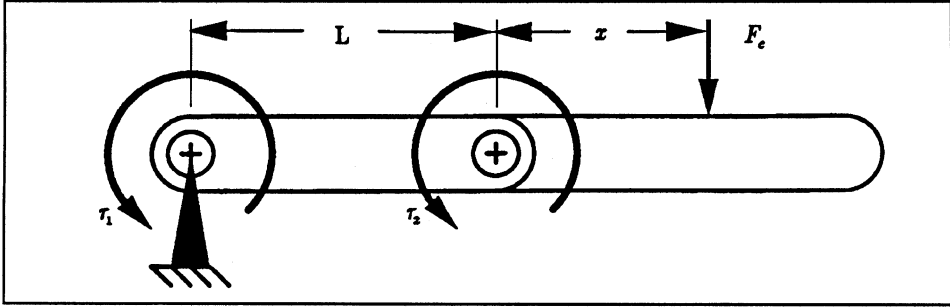


Figure 1: Determining force location and magnitude.

and τ_2 are applied (and known) by the controller, and L is the length of the inner link. The equilibrium equations of the two joint torques provide two independent equations for solution of the location of a single contact, x , and a perpendicularly applied contact force, F_c .

$$\dot{F}_c = \frac{\tau_1 - \tau_2}{L} \quad (1)$$

and

$$x = \frac{\tau_2 L}{\tau_1 - \tau_2} \quad (2)$$

With four independent degrees of freedom we can determine the magnitude and line of action of a single contact force on the last link.

2 Special Requirements for Whole-Arm Manipulation

Whole-arm manipulation tasks require a re-evaluation of the performance requirements which guide the design of robotic hardware. This section describes five requirements which helped guide the design of the WAM arm.

2.1 Large Dynamic Range of Force Controllability

The dynamic range of force controllability is the maximum controllable force (strength) divided by the minimum controllable force (accuracy) of the manipulator at a single point of contact. This ratio is useful in maximizing the range of tasks a particular manipulator can perform. One can increase without bound the maximum controllable force of a manipulator by giving it higher-torque actuators and bigger, stronger links. Similarly, one can achieve very small controllable forces by building a small, light manipulator with lightly preloaded bearings and sensitive force sensors. However, intricate tasks often demand the application and sensing of a broad range of forces.

Dynamic range of force control is selected in place of either strength or accuracy in order to address task performance more directly. This measure is dimensionless and independent of scale.

Since dynamic range is limited by the maximum and minimum controllable forces, we examine each of these separately. The maximum controllable force is limited by the motor-torque saturation limit times the transmission ratio, the strength of the transmission, and

the strength of the links. In a system without explicit force feedback, the minimum controllable force is limited by torque ripple, dry friction, and deadband in the motor controller. In a system with force feedback, limit cycles arising from the combination of highly nonlinear elements and feedback control often limit the minimum controllable force.

2.2 Robustness

The manipulator must be robust. We attach a broader definition to the word “robust” than do researchers of systems and controls. By robust we mean that the manipulator be able to perform its tasks reliably and without suffering damage. A robust manipulator then must be

- dynamically stable and
- mechanically durable
under all conditions.

Dynamic Stability:

Maintaining dynamic stability is the type of robustness most commonly referred to in systems and controls. This sense of robustness requires that a dynamically stable manipulator remains dynamically stable in the face of changing inputs, disturbances, payload, contact stiffness, and arm configuration. Advanced techniques such as sliding-mode control have been developed which deal directly with parameter uncertainty [Slotine 84].

Mechanical Durability:

To maximize mechanical durability, on the other hand, we want to minimize any impact-induced force and to minimize fragility so that the manipulator can bash around, exploring an uncertain environment without damaging itself. We can improve the survivability of the manipulator by using tough materials, tucking fragile transmissions and sensing mechanisms inside the load-carrying structure, adding protective coverings around the links and joints, and minimizing the forces of impact.

By equating kinetic energy before collision to potential energy during collision, we find that the maximum impact force, $F_{c_{impact}}$, is

$$F_{c_{impact}} = v_l \sqrt{J_l k_c} \quad (3)$$

where

J_l is the inertia of the link measured at the point of contact,

k_c is the contact stiffness, and

v_l is the contact velocity.

In order to minimize impact forces, the designer must minimize the backdriven mass, increase the contact compliance (perhaps with a soft covering), and limit velocity of the moving mass.

2.3 High Bandwidth

High bandwidth of force and position control is important in manipulators used for assembly, where the cycle time of tasks is critical. Bandwidth is also important for controlling

forces against shaking and undulating environments. For position-controlled manipulator designs, maintaining high bandwidth may be an unconscious decision; but, since some designers believe that force-controlled manipulators should be naturally compliant [Nevins 73, Andeen 88], the danger exists for their resulting designs to exhibit significantly lower mechanical bandwidth.

2.4 High Aspect Ratio

We define aspect ratio of the link as its length, L , divided by its width, W . When the aspect ratios are high, the links are long and slender. In all manipulators, increasing the aspect ratio increases the unobstructed workspace and allows the manipulator links to reach in and around obstacles in the environment more easily.

High-aspect-ratio links are better at grasping and manipulating as shown in figure 2. This figure illustrates a serial-link manipulator trying to grasp a cylindrical object of diameter, D_{cyl} , between consecutive links of length, L . In each case the link width, W , and the coefficient of friction, μ , between the object and the cylinder, are the same and determine the friction-cone angle and the maximum permissible joint angle, each equal to θ , which allows a secure grasp. The useful grasping length, L' , normalized by the link length, L , is

$$\frac{L'}{L} = 1 - 1/2 \frac{W}{L} \frac{1}{\tan \frac{\theta}{2}}, \quad (4)$$

and the largest cylinder that the pair of links can grasp is

$$\frac{D_{cyl}}{L} = 2\mu - \frac{W}{L}. \quad (5)$$

Therefore, the grasp length, L' , and the largest diameter cylinder that can be grasped are dependent directly on the aspect ratio, $\frac{L}{W}$, and are both maximized when the aspect ratio is made as large as possible.

Many design considerations affect aspect ratio. For example, the decision to use a compact transmission in order to remove actuator bulk from the joint to the base or just back a few links improves aspect ratio. When a not-so-compact, single-stage reduction is used, the diameter of the final drive pulley or gear at the joint or the length of the output link of a four-bar linkage, if made large to increase the effective transmission stiffness, transmission ratio, and/or joint strength, decreases aspect ratio. When a more-compact multiple-stage reduction increases torque at the joint, aspect ratio is improved over the single-stage reduction at the cost of lower power efficiency [Townsend 88B] and higher complexity.

2.5 Good Backdrivability

There are two types of backdrivability:

- acceleration-dependent and
- velocity-dependent.

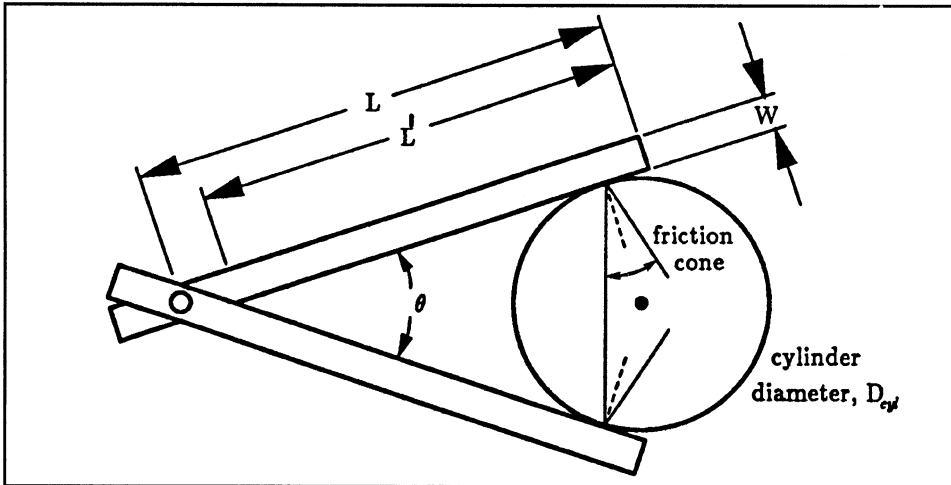


Figure 2: Effect of aspect ratio on available contact length.

Backdrivability is measured in Cartesian coordinates at a fixed point on the manipulator, usually the endtip. A mechanism which has good acceleration-dependent backdrivability generates only small inertia-induced contact forces when accelerated by the contact. The backdrivability of a single link is improved by minimizing the link-structure inertia and then keeping the transmission ratio smaller than the matched-inertia transmission ratio so that the reflected motor inertia remains relatively small.

Similarly, a mechanism which has good velocity-dependent backdrivability generates small friction-induced forces in response to imposed endtip velocities. It is commonly known that a transmission mechanism which uses worm gears and has dry friction will not be backdrivable at all if the pitch angle of the worm gear is less than the friction-cone angle. It is worth noting here that, if only power efficiencies are available to compare the quality of competing transmission designs, then in many cases the highest-efficiency drive will provide the best velocity-dependent backdrivability.

The concept of designing manipulators for good backdrivability is borrowed from high-quality teleoperator design where high backdriven inertia and friction in the master/slave system would mask the transmission of forces in bilateral force reflection. Also, isotropy in the backdriven inertia and friction improves teleoperator performance by reducing the disparity between the desired and achieved motions. Some manipulator designers as well have begun to design for good backdrivability. For example, in order to simplify the dynamic equations for calculating the actuator torques in the trajectory control of a direct-drive arm, Asada [Asada 84] redesigned a manipulator so that the inertia properties at its endtip would be nearly isotropic over a large portion of its workspace.

Good backdrivability causes the manipulator to behave desirably without dependence on closed-loop control. If closed-loop control is used, system accuracy can be improved. If open-loop force control is used and the manipulator is backdrivable over a practical bandwidth, then forces which are applied to the manipulator are "sensed" at the actuator without the need for endtip sensors. In effect, the distinction between actuator and sensor

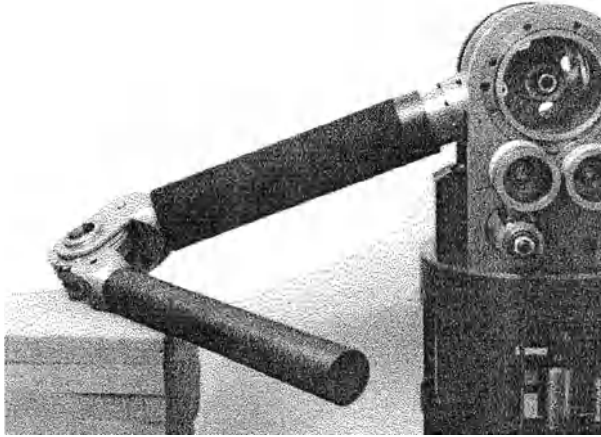


Figure 3: WAM arm exploring foam blocks.

vanish. Furthermore, the good backdrivability means that the impulse force generated upon impacts will be smaller and a manipulator will be naturally robust to collisions and impacts by lowering the effective backdriven mass of the link, J_l of equation 3.

3 Description of the WAM Design

This section describes the design of MIT's WAM arm and the results of initial experiments.

3.1 Mechanical Design

Figure 3 shows a photograph of the WAM arm. It has four revolute degrees of freedom (without a wrist): three intersecting joints at the base and one distal joint located 0.6 meters from the base. The two non-zero-length, cylindrical links have a combined reach of 1 meter and mass of 4 kg (including the distal joint). The mass of the arm including base and motors is 35 kg. All four joint axes have a range of three-quarters of a revolution.

The arm uses stiff, backdrivable, multi-stage, cabled transmissions between the compact joints and the four Moog brushless DC motors located in the base. Joint positions are inferred from 12-bit-resolution resolvers mounted on each motor shaft. Torques are inferred from Hall-effect sensors which measure the motor-winding currents. By measuring positions and torques at the motors the stability problems associated with noncolocated systems are avoided.

The transmission reducers are located directly at the joints they drive, for maximum stiffness, as explained later in this section, instead the standard practice of placing them at the motor in the form of a gearhead. The transmission ratios are sized so that the mechanical advantage is large but so that the backdriven motor inertia is negligible compared to a 0.1-kg payload. Special split-pulley designs allow single-point pretensioning with a pretensioning-propagation scheme which automatically sets the correct pretension in all stages. A novel cabled differential allows the actuators to be placed closer to the base while maintaining backlashless, efficient, and stiff mechanical-power transmission.

Even where volume would have permitted direct-drive motors, such as in the first-axis drive, relatively small motors with speed-reducing cable transmissions were used to improve torque ripple, backdriven inertia, and the effective motor constant while reducing cost and weight.

The links themselves are long and slender and are covered with a 3-mm-thick dense foam to tailor the contact characteristics for manipulating objects. Both links are tubular: the inner, aluminum link is designed to absorb large local impacts with its 5-mm thickness and small 38-mm radius of curvature (cylinder radius); the outer, carbon-fiber/epoxy link is more tailored for low mass while its 25-mm radius of curvature, 2-mm thickness, and foam covering afford it ample toughness for impacts. In order to meet these toughness constraints, the arm is many times stronger than it must be to lift its maximum 2-kg mass payload against gravity and many times stiffer than appropriate backdrivable servomotors and servomotors with transmissions can be. The links themselves are modular so that, for example, the outer link could be replaced by an actuator pack to drive a wrist and gripper.

A channel is provided for instrumentation and additional pneumatic or electric power line routing from the base to the endtip of the distal link. The manipulator's mounting is simple and requires two floor-mounted anchor bolts. The motor power supplies and analog current amplifiers are located 3-meters away in a separate electronics box.

3.2 Experimental Results

Initial experimental results from several hundred hours of tests have been reported by researchers [Niemeyer 89, Salisbury 89] at MIT's Artificial Intelligence Laboratory and Nonlinear Systems Laboratory. All of the experiments were performed without a wrist, end-effector, or other payload. In free trajectories, by using adaptive control, endtip speeds have exceeded 10.8 meters/sec with accelerations of 134 meter/sec² (13.7 g's). Using only the four motor resolvers to estimate position, the endpoint repeatability of the arm is ± 1 mm. When applying contact forces, the maximum vertical force which has been applied by the arm when outstretched horizontally is 8 kg_f. The force resolution with motor-torque ripple and cogging compensation is ± 0.2 kg_f. The stiffness of the fourth joint, which has the longest transmission span (0.6 meters), is 1800 newton-meter/sec with the motor mechanically locked. The highest controlled stiffness of the fourth joint is 100 newton-meter/sec, limited by the maximum stable motor-controller gain.

4 Novel Transmission Mechanisms for Whole-Arm Manipulation

This section describes two cabled transmission mechanisms developed specifically for the WAM arm.

4.1 The Choice of Cable Drives

When properly designed, cable drives have high material strength, low weight, low velocity and torque ripple, no backlash, and low friction. Furthermore, they do not leak, do not require surface lubrication, and can be guided over long distances around pulleys through

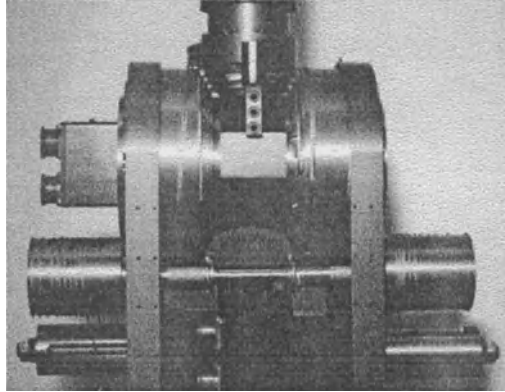


Figure 4: Differential mechanism on the WAM arm.

complex and twisting geometries. Cables and all other tension-element drives, such as tapes and belts, do not transfer power through compression or shear, and so avoid added compliance and strength limitations from bending moments or buckling. When designed for reliability, cable drives have a history of dependability in such demanding applications as aerial trams, ski lifts, cable cars, light-aircraft control surfaces, cranes, and elevators.

4.2 Cabled Differential

Figure 4 is a photograph of a cabled differential (which was invented for the design of this arm) as integrated in the design of the WAM arm. Figure 5 shows details of the cable-differential concept. Unlike traditional bevel-gear differentials, the cabled differential has only rolling contact (cable-to-pulley) and so has extremely low friction without the need for surface lubricants. The tooth-frequency torque ripple, gear noise, and backlash normally associated with traditional differential drives are eliminated. Also, fabrication is simple, requiring only concentric steps to be lathed without re-chucking the workpiece. Finally, the design is also extremely stiff because there is zero free length of cable as it unwinds from one pulley and immediately winds onto its mating pulley.

4.3 High-Speed Transmission with Specially Designed Speed Reducer

Commonly a speed-reducer mechanism is included in the transmission to boost the actuator torque capacity and the effective motor constant of a small-but-fast actuator by the magnitude of the speed reduction. Conventionally, the reducer is located at the motor shaft and the mechanical designer only selects its magnitude, N , (also called transmission ratio) which is the ratio of motor speed to joint speed. However, the designer is free to select both its location and its magnitude! Figure 6 shows the model of a transmission with a speed reducer located arbitrarily between the motor and the joint it drives. Let the distance between the motor and joint be L and the distance between the motor and the reducer be C . The cable is sized for a given maximum stress so that an N -times heavier cable is required in the low-speed tension-element.

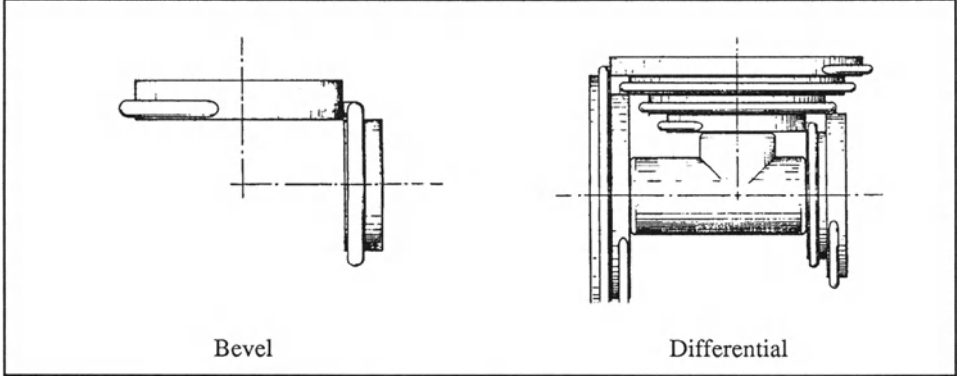


Figure 5: Differential cabling concept.

In many types of transmissions, such as tension-element drives, both stiffness and strength are proportional to the tension (or compression) cross-sectional area. Suppose the cross-sectional areas for the high- and low-speed parts of the transmission are selected so that the stress is constant along the transmission. We find that, although the element in the high-speed part of the transmission is lighter than that in the low-speed part by N , the effective stiffness of the high-speed part for a given transmission length is greater by N . Furthermore, the effective transmission stiffness, k_{eff} , for the transmission model of figure 6, measured at the joint, is

$$k_{eff} = \frac{N^2 EA}{NL + C(1 - N)}, \quad (6)$$

where L is the transmission length, E is the modulus of elasticity of the transmission-element material, and A is the cross-sectional area of the high-speed part of the transmission. The effective transmission stiffness of equation (4.1) is maximized by letting $C = L$, i.e., by placing the reduction mechanism at the joint, so that the high-speed part of the transmission spans the entire distance between the actuator and joint.

The benefits of placing the reducer at the joint would be lost if we used a geared reducer that was significantly more bulky or more heavy than the joint alone. For this reason, we developed the compact, light-weight, cabled speed-reducer shown integrated with the fourth joint of the WAM arm in figure 7. The mechanics of this design are illustrated in figure 8.

Placing the reducer at the joint has several benefits. First, since the stiffness is greater, trajectory and contact force bandwidths are increased. The greater stiffness also improves closed-loop force-control stability [Townsend 87]; and, because the transmission operates with higher tension-element speeds, the power efficiency [Townsend 88B], and therefore the backdrivability, are improved.

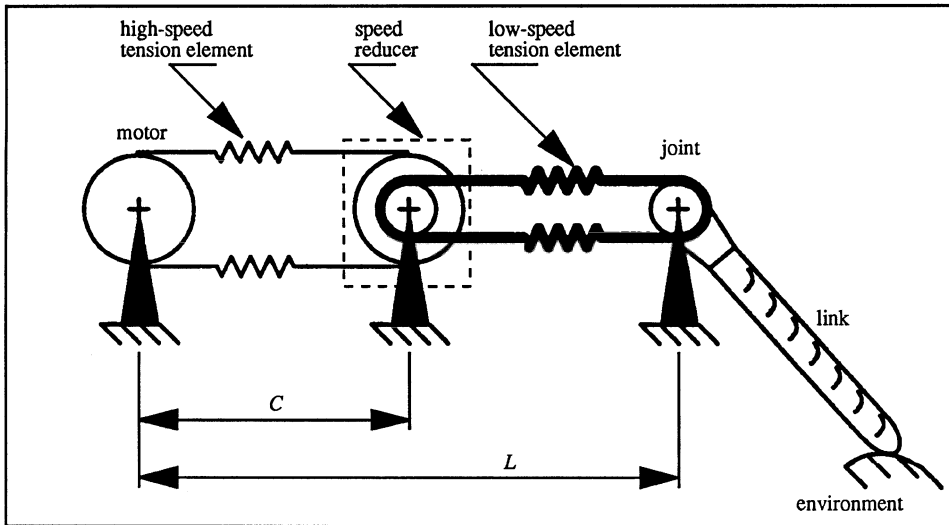


Figure 6: Choosing the speed-reducer location.

5 The Future

Although we are presently focusing on the next mechanical-design challenges of whole-arm manipulation, such as the torque-output quality of the drive motors and the integration of a WAM-style wrist and hand, it is important to consider the broader implications of whole-arm manipulation.

Research is needed to explore the new possibilities for intelligent manipulation. Since high-performance transmissions, driven by actuators of comparable quality, reduce the reliance on endtip-force feedback, high-level force-control schemes based on a vector of joint torques become practical. New strategies such as controlling large forces near the base of a link and controlling small-but-accurate forces near the tip of a link should be considered. Similar strategies could be employed to vary the effective contact impedance passively by choosing the point of contact along a link.

Appropriate kinematics must be considered to maximize the observability and controllability of forces for exploring uncertain environments and for performing a variety of manipulation tasks such as grasping between adjacent links and controlling line contacts. There are many human tasks which involve capabilities only now becoming achievable in robot hardware. In soccer, the ball is guided by a contact between it and the edge of the player's shoe. A person gathers a scattered collection of objects between her arms. The Olympic-style wrestler controls forces with tremendous speed and agility along many parts of his body simultaneously. Without vision, the edges of a box are explored quickly by groping with arms and hands. The capabilities which enable these human tasks will also enable tasks more appropriate for robotic hardware. As researchers and designers, we must expand our expectations for force control and begin to explore these entirely new horizons as robots interact more with their environment.

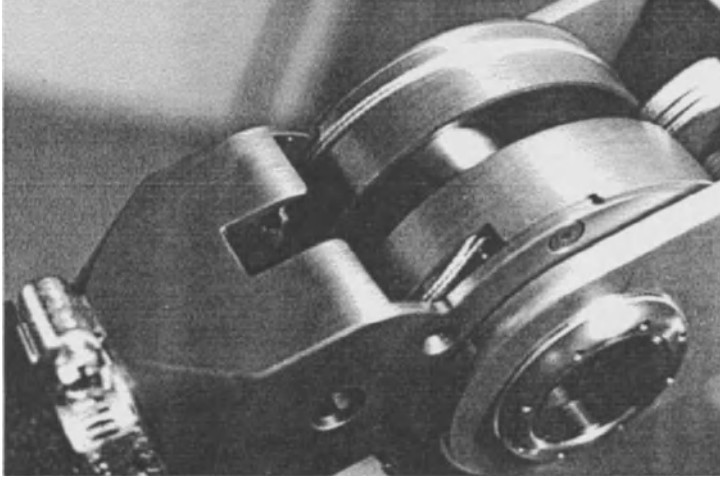


Figure 7: Speed reducer located at the fourth joint of the Whole-Arm Manipulator.

References

- [1] Andeen, G.B., and Kornbluh, R., "Design of Compliance in Robotics," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, PA, April 1988.
- [2] Asada, H., "Dynamic Analysis and Design of Robot Manipulators Using Inertia Ellipsoids," *Proceedings of the First International Conference on Robotics*, Atlanta, GA, March 1984.
- [3] Nevins, J.L., and Whitney, D.E., "The Force Vector Assembler Concept," *Proceedings of the First CISM-IFTOMM Symposium*, Udine, Italy, 5-8 September 1973, pp 273-288.
- [4] Niemeyer, G., and Slotine, J.J.E., "Computational Algorithms for Adaptive Compliant Motion," *Proceedings of the 1989 IEEE International Conference on Robotics and Automation* Scottsdale, AZ, May 1989.
- [5] Salisbury, J.K., "Teleoperator Hand Design Issues," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 1986.
- [6] Salisbury, J.K., "Whole-Arm Manipulation," *Proceedings of the 4th International Symposium of Robotics Research* Santa Cruz, CA, August 1987.
- [7] Salisbury, J.K., Townsend, W.T., Eberman, B.S., DiPietro, D., "Preliminary Design of a Whole-Arm Manipulation System (WAMS)," *Proceedings of the IEEE International Conference on Robotics and Automation*, Philadelphia, PA, April 1988.
- [8] Salisbury, J.K., Eberman, B.S., Townsend, W.T., and Levan, M.D., "Design and Control of an Experimental Whole-Arm Manipulator," *Proceedings of the 1989 International Symposium on Robotics Research*, Tokyo, Japan, 1989.

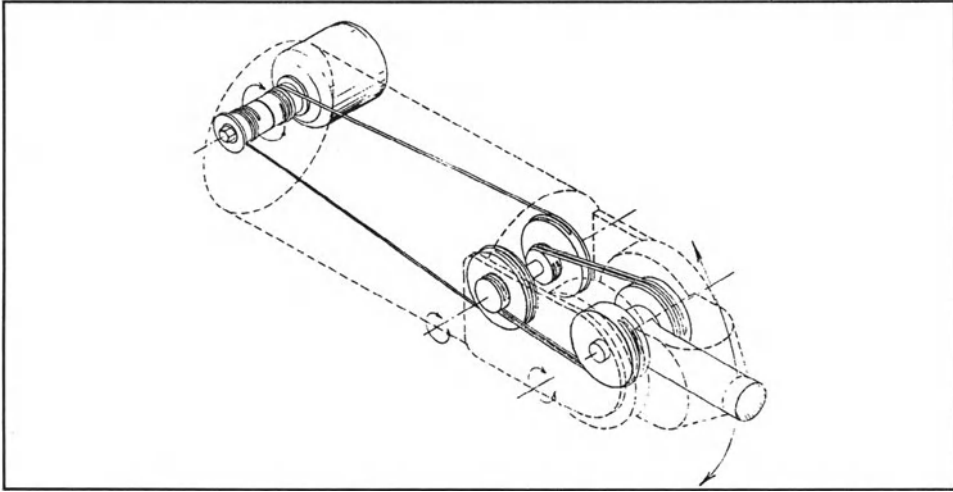


Figure 8: Mechanics of the remotely-located speed-reducer design.

- [9] Slotine, J.J., "Sliding Controller Design for Nonlinear Systems," *International Journal of Control*, Vol. 40, No. 2, 1984.
- [10] Townsend, W.T., and Salisbury, J.K., "The Effect of Coulomb Friction and Stiction on Force Control," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, Raleigh, NC, April 1987, pp pp. 883-889.
- [11] Townsend, W.T., "The Effect of Transmission Design on Force-Controlled Manipulator Performance," *MIT PhD thesis published as AI-TR-1054 by the Artificial Intelligence Laboratory*, Cambridge, MA, April 1988.
- [12] Townsend, W.T., and Salisbury, J.K., "The Efficiency Limit of Belt and Cable Drives," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 110, No. 3, September 1988, pp 303-307.
- [13] West, H., "Kinematic Analysis fo the Design and Control of Braced Manipulators," *MIT PhD thesis, Department of Mechanical Engineering*, Cambridge, MA, June 1987.

Whole-Hand Manipulation: Design of an Articulated Hand Exploiting All Its Parts to Increase Dexterity

Gabriele Vassura* and Antonio Bicchi **

* DIEM - Dipartimento di Ingegneria delle Costruzioni Meccaniche, Nucleari, Aeronautiche e di Metallurgia, Università di Bologna, Viale Risorgimento, 2, 40136 Bologna, Italy

** Centro "E. Piaggio", Facoltà di Ingegneria, Università di Pisa, Via Diotisalvi, 2 56100 Pisa, Italy.

Abstract

It is a common observation that the human hand performs various manipulation tasks using not only its fingertips, but all the surfaces available for contact, i.e. intermediate phalanges and the palm. In the first part of this paper some such whole-hand operations are discussed, relating to different domains of fine manipulation such as grasping, exploration and micro-motion of objects.

The design guide-lines deduced by the analysis of whole-hand manipulation operations in humans are outdrawn in order to reproduce a similar behaviour in a robotic hand: requirements on mechanical architecture (to provide proper surface conformation and opposition of hand elements) and on sensory equipment (to allow the synthesis of satisfactory control procedures) result from this analysis.

A second part of the paper describes how these issues can be implemented in the version II U.B. Hand, currently under development. The propensity of version I kinematic architecture to whole-hand manipulation is exploited by integrating in the mechanical structure purposely designed force/torque sensors, according to the intrinsic tactile sensing concept: the external surface of each phalange in the fingers and that of the palm thus become integral parts of as many sensing devices.

The final part of the paper provides preliminary suggestions on how to use the proposed hand to perform some tasks requiring whole-hand manipulation.

Introduction

It can be observed, in many robotic applications, that the potential functionality of existing robotic devices is seldom fully exploited, often due to limitations in sensory equipment or control procedures, but sometimes also to limits in their original conception.

As an example, most present robots are designed to interact with the environment through their end-effector, thus limiting the range of possible operations and objects the robot can deal with; the *whole arm manipulation* concept, involving the use of most parts of the robot arm to accomplish an enlarged set of tasks, was only recently proposed by [Salisbury,87], and preliminary applications are being presently demonstrated.

In the field of articulated robot hands, the one this paper is concerned with, a parallel can be easily drawn with the above example: most present robot hands are designed (or at least are used as if designed) for manipulating objects using only their fingertips, while the human hand performs various manipulation tasks using all the surfaces available for contact, i.e.

fingertips, intermediate phalanges and the palm. This results in more powerful grasps, finer control of object motion, or better sensory information, which leads to enhanced dexterity of the hand.

In practice, the performances of some articulated hands, in spite of their complex and expensive multi-dof mechanical structure, are not so far from those of simpler and cheaper grippers. It is authors' opinion that some design criteria need to be revised in view of more effective mechanical and sensory equipment integration and that useful results can be obtained if the means for full exploitation of hand elements are provided in the design phase.

Borrowing the term from Salisbury, by *whole hand manipulation* (WHM) we mean that all the links of the multi-D.O.F. kinematic chain of the hand can be used to contact and sense the object. As in the whole arm manipulation case, the WHM concept has been derived from observation of a biological system, the human hand, but design solutions are not necessarily anthropomorphic.

The goal of the work reported in this paper is to realize an artificial hand that can perform some WHM operations. In order to do this, three main aspects have to be developed: i) the hand design must allow for suitable kinematics, ensuring proper mobility and opposability of hand's elements; ii) sensors must be integrated in all the parts of the hand that are used to contact manipulated objects, and iii) sensory control methods have to be developed to guarantee the necessary degrees of flexibility and adaptability to unpredictable environments. Although the main stress of the paper is on the design of the mechanical and sensory components, other aspects of the project will be addressed.

The implementation of the concepts discussed in this paper is being currently carried out: a prototype finger, suitable for whole hand manipulation, has been built and is described in this report. Previous experience with designing and testing the version I UB Hand, providing the basis for the mechanical arrangement of the newly proposed one, will be also briefly explained.

Examples of whole hand manipulation in human activity

The functionality of the human hand has been widely investigated in its various aspects; [Schlesinger,1919] [Keller,1947] [Tubiana,1981]; it is however intuitive to verify how frequently each part of the hand (the palm, the phalanges and the fingertips) gets into contact with the objects. In the following, some cases of human whole hand manipulation are commented on in order to extract suggestions for robotic hand design.

The first example (see Fig. 1a,b,c,d) refers to a typical pick and place task for objects of similar shape (a rectangular prism) but different size and mass. Different grasp configurations, each one involving more contacts of larger area, are used by the human hand in order to improve stability.

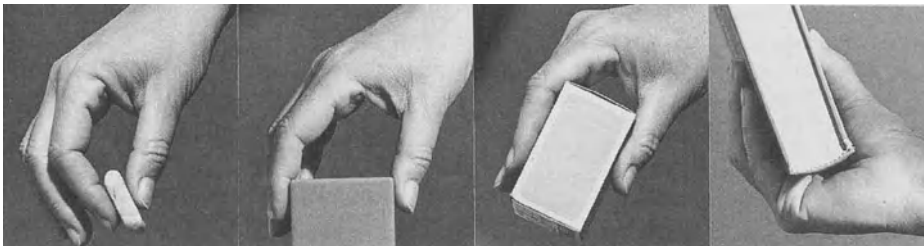


Fig. 1 Four grasps with different extension of contact surface

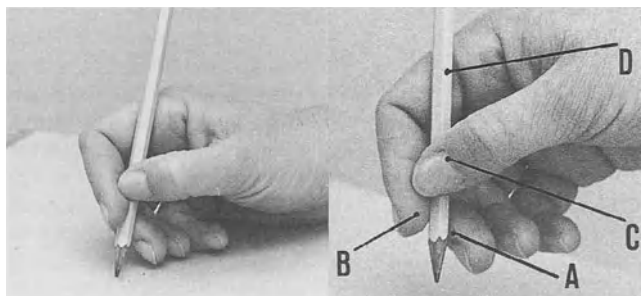
The progressive involvement of further structural elements of the hand (the inner phalanges and the palm) leads to increased strength of the grasp against the weight of the body being lifted.

The second example relates to a task which requires a "power grasp" of a tool in a constraining environment. The tool must be initially grasped in a configuration which is compatible with the constraints (the hammer is lying on a plane), and then moved inside the hand towards a final grasp configuration which is suitable for the task accomplishment. This case is very common when operating in an unstructured environment, where many objects have limited accessibility for grasping; this fact imposes initial grasp configurations different from those required by the task. Another typical example is picking up a pencil in fingertip prehension, and then manipulating it to the final configuration of fig.3. The hand operates a first grasp acting on the available surface of the object, typically in fingertip mode (fig. 2a), then partially lifts it, while the object is forced to move through a number of intermediate configurations by controlled slipping or rolling or by fingers relocation (Fig. 2b). Once the final configuration (Fig. 2c) has been reached, the power grasp of the tool and the task accomplishment become possible. This example shows how the whole surface of the hand is used not only in final constraining, but is also crucial to implement internal manipulation procedures.



Fig. 2 Manipulation before final grasp

Finally, a third example relates to a fine manipulation task, which consists of holding a pen and writing. The pen, Fig.3, is usually held in a four contact grasp with a lateral contact on the medium fingertip (A), two contacts on the index finger, fingertip pad (B) and lateral surface of the proximal phalange (D), one contact on the thumb fingertip (C). The task of writing along a line is a combination of transverse motion of the pen tip, obtained by fine motion of fingers, and line motion of the hand, achieved by moving the wrist or even the arm. It is interesting to note that in points A, B, C no slippage or rolling usually occur and small



motion between the fingers and the pen is allowed by the compliance of pads, while in contact D slippage is frequent. It is also relevant to note that the stability and the precision of the grasp greatly depend on using contacts on the lateral surface of fingers.

The observation of the biological model proposes useful suggestions and encourages to implement them in robotic hands design. In the following section, some consequent design issues will be presented.

Design issues and requirements

In order to reproduce some of the capabilities of the human hand, including whole hand manipulation, a robotic system must exhibit a number of features inherent to its mechanical design as well as to its sensory equipment, control methods and computational architecture. In this section, some of these requirements are examined, trying to make the suggestions coming from the human model explicit.

-Number of fingers. The minimum number of frictional contacts necessary to firmly grasp a generic object is three [Salisbury,82]; therefore, to obtain general enough grasp capabilities, three fingers are the least possible number. With three properly designed and controlled fingers is also possible to move the grasped object in all directions and orientations. A fourth finger is useful (but not strictly necessary) in some cases, e.g. when an exploration of the surface of the object being grasped is required. A fourth or even a fifth finger are useful to augment the strength of the grasp in heavy tasks.

-Number of DOF's. The minimum number of independently actuated joints in the fingers to obtain full mobility of the grasped object is three, if slip motions between the finger pads and the object surface are allowed [Salisbury,82]. On the other hand, if the capability of rolling the fingertips relative to the object is desired, then at least three parallel joints per finger are necessary.

-Opposability of fingers and palm. Besides by increasing the internal mobility of the hand, manipulation dexterity can take advantage by proper configuration of the DOF's in the kinematic chain. From this point of view, the position and the excursion of each joint can greatly affect the resultant manipulability [Kerr,86]: e.g., the ability of the "thumb" to rotate about an axis normal to the palm surface permits the opposability of the lateral surfaces of the "index" fingers.

-Shape of the phalanges. The smoothness of the surfaces of the hand links plays a key role in allowing controlled fine motions of an object, obtainable by rolling and/or slipping. As taught by biological models, an elliptic or circular phalange cross section often provides well shaped contact areas for bodies of any shape. Another important requirement is related to the smoothness of surface connections between adjacent links: a conical or cylindrical shaping of the whole finger allows in the possibility to easily move the contact point from one link to another during manipulation and to extend contact area to more than one link when operating with large, flat objects.

-Material properties of the pads. Some characteristics of the finger surface are desirable for dextrous manipulation. High friction, low stiction, and rather compliant materials can greatly increase grasp stability, by extending the effective contact surface with smooth objects or by reducing edge effects when sharp bodies are manipulated.

-Proprioceptive sensory equipment. Sensing the internal variables of the hand is necessary to realize effective low-level control loops of actuators. In particular, joint position sensors must have high resolution to allow fine control of finger motions. Joint torque sensors can be used to close a control loop around the disturbance source (mainly friction in mechanical transmission of power from actuators to the joint), thus achieving better control performance; however, these sensors are not necessary if good transmission means are adopted.

-Exteroceptive sensory equipment. The role of exteroceptive (i.e., relating to interactions with the environment) sensory information in dextrous manipulation has been widely recognized ever since a relevant literature appeared in this field. Indeed, their importance results intuitively when considering how the human hand can perform innumerable tasks effectively. Notwithstanding this, the analysis of functional requirements for the exteroceptive sensory equipment of a dextrous hand has not yet been carried out satisfactorily. The fundamental work of Harmon (see e.g. [Harmon,82]) in the field of tactile sensing, for instance, consisted in reporting what a group of industrial and academic researchers felt to be the necessary features of tactile sensors. These opinions, though influential, derived from the assumption of biomorphic models for sensors more than from objective functional analysis. The development of analytical methods for dextrous manipulation and the appearance of innovative non-anthropomorphic contact sensors offered a new viewpoint for the statement of sensory requirements (for an introduction of these themes, see [Mason,85]). In the following text we will briefly discuss some functional considerations on the sensory equipment of dextrous hands, with particular reference to whole hand manipulation. A subdivision of dextrous manipulation tasks in three main classes will be considered: micro-motion, grasp, and exploration of manipulated objects.

a) Micro-motion. The accomplishment of fine motion of objects held by an articulated hand necessitates of three basic steps: i)determination of the kinematic relationship between object motions in cartesian space and motions of the contact points between the object and the hand phalanges and palm (i.e., identification of the grip transform, see [Mason,85]); ii)determination of the kinematic relation between motions of contact points and motions of hand joints (i.e., the hand Jacobian); iii)control of joint position along specified trajectories. For a precise determination of both the grip transform and the hand Jacobian, the accurate knowledge of contact points is mandatory. Hence, sensors in the hand must primarily provide information about the position of every zone of contact between the object and any element of the hand (finger phalanges and palm).

b) Grasp. The grasp of an articulated hand on an object can be described by the number and position of finger-object contacts and by the wrenches exerted through the contacts. In order to synthesize a grasp, the hand controller has to determine i)where to put the finger phalanges and the palm with respect to the object surface, and ii)the intensity and direction of the wrench in each contact. The former is basically a planning problem, which can be approached on the base of an a priori knowledge of the object shape (see e.g. [Nguyen,86]) or with the help of global sensors like vision. The choice of optimal contact wrenches can be carried out, in the assumption that grasp geometry and external load is exactly defined, by criteria as those proposed by [Kerr,86] and [Bologni,88]; an adaptive method for choosing grasp forces in changing conditions has been proposed by [Bicchi,89]. Effective control of contact wrenches requires a sensor to feedback (besides contact positions) the 3-component vector of contact force and the 3-component vector of contact torque, where contact force and torque mean the resultants of distributed pressures over the contact area. Slippage avoidance (or control) is also a major concern in object grasping: a sensor able to evaluate slippage danger at each contact, and to detect when slippage actually occurs, would be very useful for grasping operations.

c) Exploration. By the use of active exploration of objects by an articulated hand, it is possible to obtain a very rich information about the object characteristics otherwise achievable with difficulty. As an example, one could manipulate the object to know its shape, the texture of its surface, its hardness, its thermal or even chemical properties, etc. Sensory equipment for achieving these information might consist of several transducers based on different principles. However, we will consider here only the features that the hand sensors must exhibit in order to allow the basic explorative movements which are prerequisite for most active perceptual tasks, i.e. to move a finger along the object surface while exerting a

controlled pressure on it. To do this, control algorithms can be developed (see [Bicchi,89]) which require information about contact position and measurement of contact forces and torques on the hand surface.

It should be pointed out that so far we referred to contact points as if contacts occurred at single points on the object and hand surfaces. This assumption is not verified when rather compliant materials are employed to cover hand's surfaces (or the object itself is compliant). In this case, a small-area contact will occur most often; information about contact area shape, and possibly about very small object features contained in such area, could be required to the sensory equipment of the hand. In most cases, though, it could suffice to know approximately the position of the contact area on the hand surface, by knowing the position of one of its points.

Robotic end effectors: an overview

The idea of using all the parts of the hand to manipulate objects is the obvious result of the observation of the human example; thus, the tendency to reproduce this capability with artificial devices can be traced back to the earliest prosthetic hands, developed several tens of years ago. Of course, the lack of any sensory and control capability prevented such devices from achieving any autonomous dexterity.

A review of the state of the art of robotic devices puts in evidence that, while some attempts have been made to design mechanical structures exploiting all their elements for some manipulation tasks, their application has been hindered again by the unsuitability of sensory equipment and by practical limitations of control algorithms. In most cases, manipulation control methods have been defined (and sometimes implemented) for multifingered hands operating with their fingertips only: significant contributions to the analysis of multifingered hand capabilities are in [Yoshikawa 85], [Kerr,86] and [Li,88]; the recent work by [Li,89] provides an elegant formalization of manipulation modes, where rolling, slipping and finger relocation are considered.

Robotic end-effectors with a palmar surface acting in opposition to the fingertips have been proposed by [Skinner,75] and [Rovetta,77], while an adaptable grasping device with many contact surfaces distributed all along the kinematic chain of each finger (resulting in an articulated tentacle) was designed by [Hirose,78]. A multifingered gripper capable of adaptable grasping with many contacts enveloping an object was proposed by [Vassura,80]. An articulated, three fingered hand proposed by [Okada,79] fulfilled some of the structural requirements to perform WHM: even if a palmar surface was not present, and only joint position sensors were employed, the finger design was suitable for locating contacts on lateral surfaces of distal and intermediate phalanges and some exhibitions in fine motion of objects were performed. The Utah-MIT Hand design [Jacobsen,84], being basically anthropomorphic, is in principle suitable for whole hand manipulation tasks, even though the sensorization and control of the device have not yet been perfected. The Stanford-JPL Hand of [Salisbury,82] rather emphasizes fingertip manipulation, being however at present capable of very fine manipulations by exploiting sensory feedback from built-in force-torque sensors located on the fingertips. Other known dexterous hands projects, as the Karlsruhe [Doll,88] and the MITI [Kaneko,88] hands, seem mainly oriented to fingertip manipulation.

The version II Belgrade hand [B&L,89] is able to grasp objects using its phalanges and palm; the variable configuration gripper designed by [Ulrich,88] emphasizes the role of the palm in order to enhance its grasping capability. No provision is made in these projects for dextrous manipulation control through sensory feedback.

An interesting implementation of a prototype hand has been presented by [Oomichi,88]: tactile and force sensors are integrated also in intermediate phalanges and the palm is exploited in power grasps.

As a conclusive remark, it can be observed that, even if a widespread opinion holds that most important research topics in the area of dexterous manipulation are related to control

algorithms and task planning, much is left to do also in hand design, since a satisfactory integration between mechanical structure and distributed sensory equipment enabling the achievement of dextrous, whole hand manipulations is still far from being completed.

The U.B. Hand

The first version of the U.B. Hand [Belletti,86] [Bologni,88] started working on a test frame in March 1988 and has been operative on a IBM 7565 gantry robot since the beginning of 1989; most of the experimental work carried out so far has consisted of examining the reliability of the proposed device and evaluating its effectiveness in dextrous manipulation tasks, with particular reference to grasping.

In order to provide a quantitative measurement of the hand grasping ability, to enable a comparison of different hand designs, and hence to guide in the choice of possible solutions, the need has been felt to overcome the limitations of previous grasp classification methods, which were found to be qualitative and insufficiently detailed. A method for the classification of all the achievable grasps has been proposed by [Bologni,88].

The proposed method is based on the generation of a table of the feasible contact configurations, where all possible oppositions of the hand elements (proximal, intermediate, and distal phalanges of the two "index" fingers with each other or with the proximal and distal phalanges of the "thumb" and the palm) are enumerated (see Fig.4).

The version I U.B. Hand has been evaluated according to this method, showing that its kinematic configuration was suitable for whole hand manipulation. Experimental tests confirmed the hand effectiveness, especially in grasping: some whole hand grasps, mimicking the above presented human hand examples, are shown in Fig.5. In order to further improve mechanical effectiveness and reliability, and to exploit the propensity of version I kinematic architecture to whole-hand manipulation, a second version of the

OPPOSITION MODE	CONTACTS											
	FINGER 1			FINGER 2			THUMB			PALM		
FINGER vs FINGER	P	I	D	P	I	D	P	I	D			
FINGER(S) vs THUMB												
FINGER(S) vs PALM												
THUMB vs PALM												
FINGER(S) & THUMB vs PALM												
FINGER(S) vs PALM & THUMB												

Fig.4 The table of opposition modes

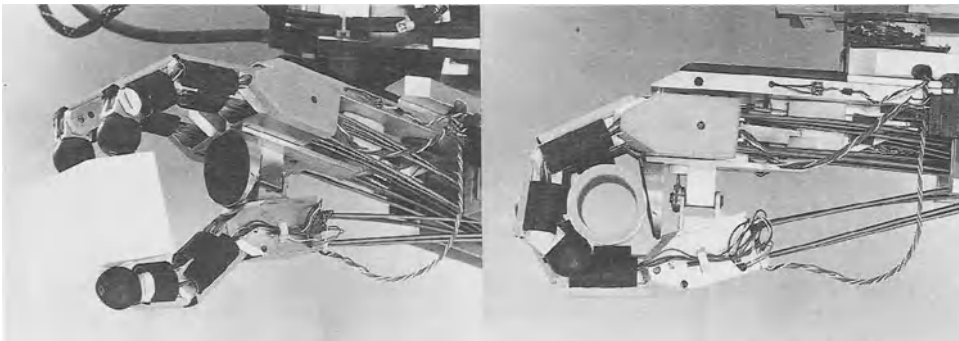


Fig.5 The U.B.Hand

hand is being developed.

The fingers of the version II UB Hand are designed according to a biomorphic skeleton-and-flesh model: in each phalange, an external shell, covered by a compliant high friction layer, and capable of sensing contacts, is connected to an inner rigid element of the kinematic chain.

The skeleton structure is composed of CNC machined links, connected through ball bearing revolute pairs. The design emphasizes modularity and tends to permanent assembly solutions in order to increase reliability and reduce the number of parts. The actuation of the 11 joints of the fingers is obtained through tendons and pulleys. The adopted configuration permits an easy removal of the external shell, so as to allow thorough accessibility and easy intervention on tendons. A detailed sketch of one of the "index" fingers of the hand is shown in Fig.6.

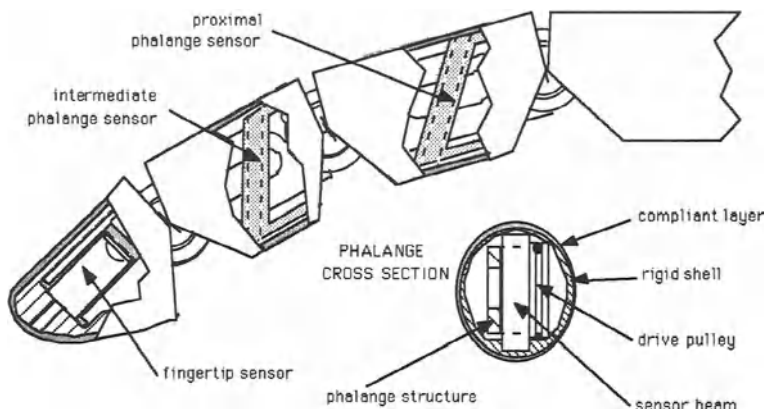


Fig.6 Integration of sensors and mechanical structure

The shape of the external shell of finger phalanges has been chosen so as to provide a regular surface for contacts all around the finger axis. The intermediate phalanges are covered with a cylindrical surface with elliptic cross-section, while the fingertip shells are revolution ellipsoids with the longitudinal axis inclined 20 degrees in the upward direction. A flat surface in the upper region enhances the approach capability, as shown in Fig.7. Finally, the palm surface has been designed as a portion of the convex surface of a large radius sphere.

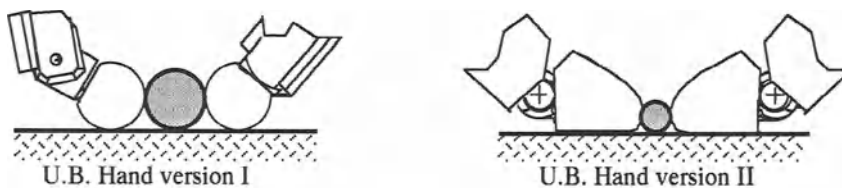


Fig. 7 Fingertip approach capability

The general view of the version II UB Hand can be seen in Fig.8: the modular design will allow the synthesis of different configurations, e.g. by varying the relative position of fingers with respect to the palm. The adduction-abduction movements of the upper fingers are independent, so that synchronous lateral movements of both fingers are allowed.

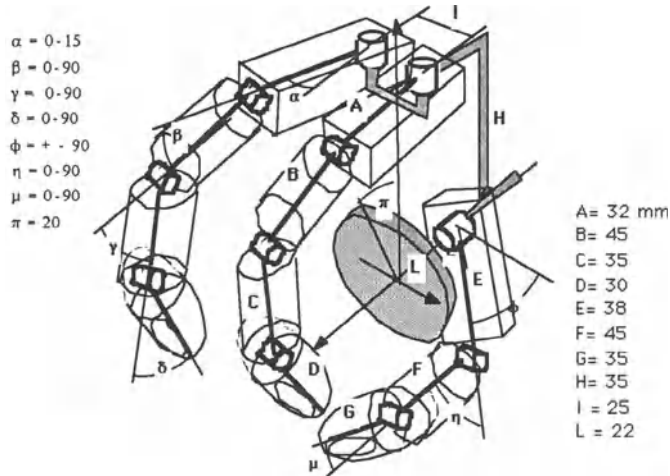


Fig. 8 Kinematic architecture of the U.B. Hand

The sensory equipment of the hand consists basically of joint position sensors (conventional shaft encoders on motor axes) and contact sensors, realized by means of the Intrinsic Tactile sensing method [Bicchi,87]. In fact, this approach seems to satisfy most of the functional specifications above examined, while its implementation does not require too complex hardware and software means.

An IT sensor is very simple in its constitutive parts, which are a 6-axis, force/torque sensor and a cover shell whose surface (the phalanx or palm surface) has known geometry.

According to the results of [Bicchi,89], if the IT sensor shell contacts an object with a small area-type contact, and adhesive forces are not exerted through the contact, by elaboration of the force/torque and geometrical information it is possible to know:

- the position of the contact centroid, that is a point on the shell surface which is assured to be internal to the contact area;
- the resultant contact force applied at the contact centroid, in intensity and direction;
- the resultant contact torque, in intensity and direction.

By comparing this with the information needed for a manipulation-oriented sensory system, it is revealed that most conditions for W.H.M. are fulfilled by IT sensing; the exception is the capability of fine imaging of features inside the contact area. We decided to postpone the realization of such fine imaging, since it would have required very high resolution skin-like tactile sensors which, if at all available at present, would represent a computational bottleneck for the whole system.

In order to accomplish the required functional capabilities of the sensory system for whole hand manipulation, IT sensing had to be realized in each phalanx and in the palm, making a total of 9 sensors.

A crucial problem in IT sensing implementation is the miniaturization of the 9 force/torque sensors employed in the hand. Different design schemes have been adopted in order to fit them in different parts of the hand, namely the fingertips, the intermediate phalanges and the palm. All sensors however employ semiconductor strain-gauges applied to deformable aluminum structures; common to the design of all the force/torque sensors is also the optimization approach employed to maximize sensor accuracy notwithstanding the small size of the sensors. This approach uses a modellization of force/torque sensors in terms of linear operations on the vector of strain measurements $\underline{V}$ obtained from strain-gauges:

$$\underline{V} = \underline{C} \underline{P} \quad (1)$$

where $\underline{C}$ is the compliance matrix of the mechanical structure of the sensor, relating the load

vector $\underline{P}$ to the measurements $\underline{V}$. The load vector $\underline{P}$ is composed of the unknown six components of the force and torque acting on the sensor in a specified reference frame; the components of $\underline{P}$ are normalized with respect to the nominal value of each component, so that the norm of $\underline{P}$, $\|\underline{P}\|$, is always less than or equal to 1.

Such modellization of the force/torque sensor leads to some considerations about sensor design: the first is that, if a 6 components load vector $\underline{P}$ is to be measured, then obviously only 6 measurements are strictly necessary. In the design of a force/torque sensor with stringent size limitations, keeping in mind this fact, though trivial, may be useful.

Numerical stability analysis techniques may be applied to the linear model of the sensor in order to evaluate its accuracy. The causes of errors in a multicomponent sensor can be in fact divided into three main groups:

i) errors in strain measurements, caused by instrumentation inaccuracies, noise etc. These errors reflect in a term $d\underline{V}$ which is summed to the measured strain vector $\underline{V}$.

ii) errors in the compliance matrix coefficients, due to the lack of exact knowledge of the load-strain relationship for the sensor structure. The $\underline{C}$ matrix can be in fact evaluated both numerically (e.g. with beam theory or with finite elements methods) and directly, by calibrating the sensor with known loads; anyway, an error matrix $d\underline{C}$ will result from modeling inaccuracies or from experimental errors.

iii) possible amplification of the errors above can occur while solving the linear system (2):

$$\underline{V} + d\underline{V} = (\underline{C} + d\underline{C}) (\underline{P} + d\underline{P}) \quad (2)$$

Equation 2 represents the true load-measurement relationship idealized in (1); $d\underline{P}$ is the error resulting on the ultimate information of the force sensor, the load vector $\underline{P}$.

In case a minimal sensor design is adopted, i.e. as many strain gauges are used as the load components are, the generalized form of Wilkinson's formula for error propagation can be applied to give an a priori estimate of the relative error on $\underline{P}$:

$$e_p = (e_v + e_c) K_p(\underline{C}) \quad (3)$$

where $e_v = \|d\underline{V}\|/\|\underline{V}\|$, $e_c = \|d\underline{C}\|/\|\underline{C}\|$, and $e_p = \|d\underline{P}\|/\|\underline{P}\|$, are respectively the relative errors on strain measurements, on calibration and on the results. The propagation factor $K_p(\underline{C})$ has an upper bound that is close to the condition number of the compliance matrix $\underline{C}$:

$$K_p \approx N(\underline{C}) = \|\underline{C}\| \|\underline{C}^{-1}\| \approx 1$$

If more strain-gauges are employed in the sensor than are strictly required, a slightly more complex propagation formula can be obtained (see [Bicchi,89]).

From this analysis of the causes of errors in force/torque sensors, it follows that possible means to increase accuracy are substantially two: a) to reduce the source errors i) and ii), by basically employing more sophisticated technologies in strain measurement and calibration, and b) to reduce the amplification of source errors by minimizing the condition number of the compliance matrix. While further source error suppression will conflict at some point with given technological or economic limitations, error propagation can be limited by carefully designing the sensor. Hence, in designing the various force/torque sensors employed in our articulated hand, we used an optimization method whose merit criterion was the minimization of the condition number of the sensor compliance matrix.

The structure of the sensors realized inside the fingertips, the intermediate phalanges and the palm are shown respectively in Fig. 9 a,b,c.

The structure of fingertip sensors simply consists of a thin walled cylinder, on which strain

gauges are applied at optimal locations and orientations. The sensor arrangement has some attractive features, which have been discussed in [Bicchi,87].

In the intermediate phalanges sensors, one end of a rectangular cross-section, internally drilled beam is fixed to the phalanx shell, the opposite end being fixed to the finger frame (the skeleton, so to speak). Gauges are bonded on the beam surface; the length of section sides, the radius of the internal hole, the position and orientation of the gauges have been chosen following the above described optimal design procedure.

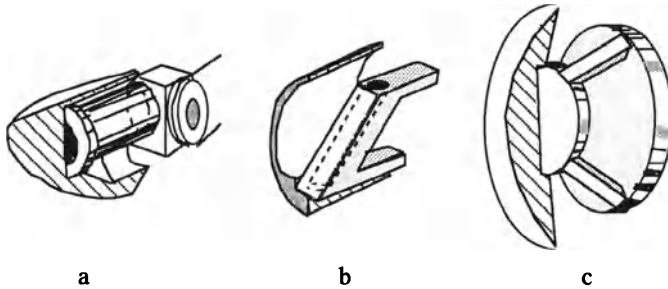


Fig. 9 Sensor configuration for fingertip, phalange and palm

Finally, the palm sensor sketched in Fig. 9c consists of three thin flexures, placed behind the palm surface, on which strain-gauges are placed. The flexures are inclined and are spaced 120 degrees apart. Again, the inclination angle, and the location of strain-gauges on the flexures are chosen to optimize sensor accuracy.

Conclusions

The paper has reported on the design issues of both mechanical and sensory equipment of a robotic articulated hand, by which tasks will be accomplished with full exploitation of the available links. Following from an analysis of some dextrous manipulation operations performed by the human hand, some design requirements have been derived: it should be possible to touch manipulated objects with every part of the hand surfaces and it should be possible to detect the position of each contact point and to measure the forces and torques exerted by contact.

The basic idea of integrated and distributed Intrinsic Tactile sensoriality in a purposely designed mechanical configuration, the version II U.B.Hand, has been presented, and its practical feasibility illustrated. The mechanical and sensory equipment design are the first step towards the implementation of a complete system for dextrous manipulation, of which some elements are still being developed. However, a prototype finger designed according to the proposed principles has been realized, and preliminarily tested.

Despite the broad potential of the device, deriving from the 11 D.O.F.'s and thorough sensorization, the resulting design appears to be reasonably compact and feasible, both for its mechanical structure, due to simplification in machining and assembly process, and for sensory equipment and data integration, due to the peculiarities of intrinsic tactile sensing.

Future work will be mainly focused on the design and implementation of suitable computational architecture and sensory control procedures.

Acknowledgements

This work reports on the activity of the interdisciplinary research group from DIEM and DEIS of the University of Bologna, composed of G.Vassura and C.Bonivento, as coordinators, and

L. Bologni, S. Caselli, E. Faldella, C. Melchiorri, V. Parenti Castelli and A. Tonielli. The version 2 UB Hand has been designed in cooperation with the Centro "E. Piaggio" of the University of Pisa, where the sensory equipment has been developed.

The authors wish to thank Stefano Monti for his activity in setting up the version II finger prototype.

Financial support was provided by IBM Italia and CNR Progetto Finalizzato Robotica grant N. 89.00561.PF.67.3.

References

B&L Engineering "The Belgrade/USC Robot Hand", Brochure, Preliminary Release, May 1989

M. Belletti, C. Bonivento, G. Vassura, "Sviluppo di Organo di Presa ad Elevata Destrezza per Robot Industriale", Convegno naz. SIRI, Milano, 1986.

A. Bicchi, P. Dario: "Intrinsic Tactile Sensing for Artificial Hands", Robotics Research, R. Bolles and B. Roth Editors, MIT Press, 1987.

A. Bicchi, J. K. Salisbury, P. Dario: "Augmentation of Grasp Robustness Using Intrinsic Tactile Sensing", Proc. 1989 IEEE Conf. on Robotics and Automation, Scottsdale, AZ, May 14-19, 1989.

A. Bicchi: "Strumenti e metodi per il controllo di mani per robot" (Methods and Devices for the Control of Robot Hands), Doctoral Dissertation, University of Bologna, 1989.

L. Bologni, "Robotic Grasping: How to Determine Contact Positions", Proc. IFAC Symp. on Robot Control (SYROCO), 1988.

L. Bologni, S. Caselli, C. Melchiorri, "Design issues of the U.B. Hand" NATO ARW on Robots with Redundancy, Salo', Italy, 1988.

C. Bonivento, S. Caselli, E. Faldella, C. Melchiorri, A. Tonielli, "Control System Design of a Dextrous hand for Industrial Robots", IFAC SYROCO '88, Karlsruhe, 1988.

T. J. Doll, H. J. Schneebeli, "The Karlsruhe Hand", IFAC SYROCO '88, Karlsruhe, 1988.

L. D. Harmon: "Automated Tactile Sensing"; Int. Jour. Robotics Research vol.1, No.2, pp.3-32, 1982.

S. Hirose, Y. Umetani, "The Development of Soft Gripper for the Versatile Robot Hand", Proc. 7th ISIR, Tokyo 1977

A. D. Keller, C. C. Taylor, V. Zahm, "Studies to Determine the Functional Requirements for Hand and Arm Prosthesis", Dept. of Eng. UCLA, 1947

S. C. Jacobsen et al, "The UTAH-MIT Dextrous Hand: Work in Progress", Int. Jour. of Robotics Research, vol. 3, n.4, 1984.

M. Kaneko, K. Tanie, "Basic Considerations on the development of a Multi-fingered Robot Hand with the Capability of Compliance Control", Proc. Int. Meeting on ADVANCES IN ROBOT KINEMATICS, Ljubljana, September 1988

J. Kerr, B. Roth: "Analysis of Multifingered Hands", Int. Jour. Robotics Research, vol.4, n.4, MIT Press, 1986.

M.T.Mason, J.K.Salisbury: "Robot Hands and the Mechanics of Manipulation", MIT Press, Cambridge, MA, 1985.

V.D. Nguyen: "The Synthesis of Stable Force-Closure Grasps", MIT-AI Lab Technical Report 905, 1986.

T. Okada, "Object Handling System for Manual Industry", IEEE Trans.on Systems, Man and Cybernetics, Vol SMC 9, pp.79-89 , 1979

T. Oomichi, T. Miyatake, A. Maekawa, T. Hayashi, "Mechanics and multiple sensory bilateral control of a fingered manipulator" Int. Conf. on Robotics Research, 1988

J. K. Salisbury, "Kinematic and Force Analysis of Articulated Hands", Ph. D. Dissertation, Stanford University, 1982.

J. K. Salisbury, J.J. Craig, "Articulated Hands: Force Control and Kinematic Issues", Int. Jour. of Robotics Research, vol. 1, n.1, 1982.

J. K. Salisbury: "Interpretation of Contact Geometries from Force Measurements"; Robotics Research, M.Brady and R.Paul Editors, MIT Press, 1984.

J. K. Salisbury, " Whole-Arm Manipulation", Proc. 4th Int. Symp. on Roboics Research, Santa Cruz, CA, August 87

G. Schlesinger, "Der Mecanische Aufbau der Kunstlichen Glieder", Ersatuglider und Arbeitshilfen, M. Borchart et Al., Springer, 1919

Z. Li, S. Sastry, "Dextrous Robot Hands: Several Important Issues", Proc. IEEE Workshop on Dextrous Robot Hands, Philadelphia, 1988.

Z. Li, J.F. Canny, S. Sastry "On Motion Planning for Dexterous Manipulation, Part I: The Problem Formulation", Proc. IEEE Int. Conf. on Robotics and Automation, Phoenix, 1989

A. Rovetta, "On Specific Problems of Design of Multi-Purpose Mechanical Hand Industrial Robots", Proc. 7th ISIR, Tokyo, 1977.

F. Skinner, "Multiple Prehension Hands for Assembly Robots", Proc. 5th ISIR, Chicago 1975

R. Tomovic, G.A.Bekey, W.J.Karlplus, "A Strategy for Grasp Synthesis with Multifingered Robot Hands", Proc. IEEE Int. Conf. on Robotics and Automation, Raleigh, 1987.

R. Tubiana, "The Architecture and Functions of the Hand", The Hand, Vol.1, W. JB. Saunders & Co., pp.19-93, 1981

N. Ulrich, R. Paul, R. Bajcsy, "A Medium-Complexity Compliant End Effector", Proc. IEEE Int. Conf. on Robotics and Automation, Philadelphia, 1988.

G. Vassura, A. Nerozzi, "Study and Experimentation of a Multifinger Gripper", Proc. 10th ISIR, Milano 1980

T. Yoshikawa, " Manipulatability of Robotic Mechanisms", Int. Jour. of Robotics Research, Vol.4, n.2, Summer 85

T. Yoshikawa, K. Nagai, "Analysis of Grasping and Manipulation by Multi-Fingered Robot Hands", Proc. IEEE Workshop on Dextrous Robot Hands, Philadelphia, 1988.

Stable Grasping and Manipulation by a Multifinger Hand with the Capability of Compliance Control

Kazuo Tanie and Makoto Kaneko

Cybernetics Division, Robotics Department
Mechanical Engineering Laboratory, MITI
Tsukuba City, Ibaraki, 305 Japan

Abstract

This paper deals with two fundamental problems concerning a multi-fingered robotic hand with the capability of compliance control. One of them relates to developing a torque sensor useful for the tendon-pulley driving system and the other was a stable grasping and manipulating problem. In order to construct the finger joint actuation in multifingered systems, a tendon-pulley driving system has normally been used. In this kind of driving system a compact joint torque sensor plays an important role in achieving compliant motions at the finger tip and in turn constructing dextrous multifingered hands. It seems, however, that a satisfactory torque sensor has not yet been developed for such a driving system. To cope with this, a Tension Differential type Torque sensor (TDT sensor) is first proposed in this paper and applied to a newly designed robotic hand with two articulated fingers in experiments. Secondly, the stable grasping and manipulation problems in the multifingered hand are addressed assuming the existence of friction at the contact area of each finger tip and the object grasped. To formulate a stable grasping condition, the stiffness matrix of an object grasped by fingers with compliance adjustable joints was introduced. By using the stiffness components, the condition was described in a simple form. The stiffness matrix was resolved to the simpler form at the tip of each finger when the hand was grasping an object. To satisfy the desired stiffness matrix at the tip, the joint stiffness matrix at each finger was adjusted. To manipulate an object, the desired trajectories of the object were converted to joint trajectories using the inverse kinematics equations, and the position reference at each joint servomechanism was adjusted according to them, keeping the stable grasping condition. A robotic hand with two articulated fingers equipped with specially designed small TDT sensors was constructed. Using the hand, various experiments were carried out and the proposed methods were confirmed.

1. Introduction

There are two topics described in this paper. Firstly, a newly designed tension difference type torque sensor suitable for tendon-pulley driving system is proposed. Secondly, Stable grasping and manipulation problems are discussed based on the compliance control method.

In order to grasp various shapes and to manipulate objects dexterously, numerous multi-fingered robot hands have been designed [3,5,7,9]. Because of the limited space in which to install actuators inside the fingers, actuators are usually placed on a location far from the driving

joints and the power is transmitted from an actuator to a finger joint through appropriate transmission devices. A tendon-pulley system is a popular device for this purpose. In general, the transmission system includes elements with low structural stiffness and friction such as tendons, gears, and conduits. This makes it difficult to realize a high quality compliance control of such a robot hand without any force (or torque) feedback loop.

In active compliance control for robot hands, force(or torque) sensing has been introduced either at the finger tip [10], or at the joints [3,4,9]. The force sensing at the finger tip requires the inverse kinematic calculation to determine the force at the joint space and results in low control speed. Alternatively, the joint torque sensing is effective to remove this drawback, because no coordinate transformation is necessary to provide the force feedback with each joint. Accordingly, joint torque sensing is probably appropriate for a high speed joint torque control.

The typical tendon-pulley driving system has agonist and antagonist tendons for a single axis joint. To get the single axis joint torque, the tension difference between both tendons must be measured. The popular technique is using two tendon-tension sensors installed on each tendon. The difference of the two sensor outputs are calculated to obtain torque information. Salisbury and Craig[9] have developed the cantilever type tendon-tension sensor having idler pulley and strain gauges. Jacobsen and others[3], Dario and Buttazzo[4] have used similar methods to obtain tendon-tensions for a four-fingered robot hand and a single finger system, respectively. Although the torque around the drive pulley can be successfully measured by taking the difference between two tendon-tensions, this approach has two disadvantages: (i) Two tendon-tension sensors are necessary for a joint in principle, while the tension itself is not the main point of interest to construct the control system. This might prevent us from constructing a hand system with few signal lines and simple configuration. (ii) The bias tendon-tension always acts on the base of the sensor beam. This will make it difficult to get a wide dynamic range and remove drift due to residual stress.

To cope with this, the authors propose the *Tension Differential type Torque sensor (TDT sensor)*, which is based on the idea that the torque around a drive pulley is proportional to the tension difference, and this can be measured directly without sensing individual tendon-tension. The working principle of this sensor will be demonstrated with some experimental results.

Regarding the stable grasping problem, in this paper a stable grasping condition described using compliance parameters of the grasping system is discussed. In order to have a practical discussion, the friction effects between a finger tip and the object grasped will be considered. There are many works concerning multifingered hands [1,2,6]. However, few researchers deal with the friction effects in grasping motion. Therefore, there are some difficulties for the analytical results to apply to a practical grasping system. In the analytical discussion, a grasping model including the friction effects is introduced. To find the grasping stability condition, the stiffness matrix of an object supported by compliant fingers is first constructed. Through the potential energy analysis, it will be found that grasping stability will be assured if the matrix is positive definite. To provide variable compliance functions, task coordinate position servos will be constructed at the tip of each finger. The loop gain of each servo system will be changed to put appropriate compliance to each axis on the task coordinate system. The manipulation is carried out by adjusting the reference value to each position servo system. Analytical results will be confirmed from the experiments using a robotic hand with two newly developed fingers.

2. Tension Differential Type Torque Sensor (TDT Sensor)

The basic principle of the TDT sensor is illustrated in Fig. 1(a). Let F_1 and F_2 be the tendon-tensions at both sides of pulley A, respectively. Then the torque T around the drive pulley is given by

$$T = r (F_1 - F_2) \quad (1)$$

where r is the radius of the pulley. Note that the torque can be obtained by directly measuring the tendon-tension difference instead of measuring two tendon-tensions individually. The TDT sensor is based on this idea.

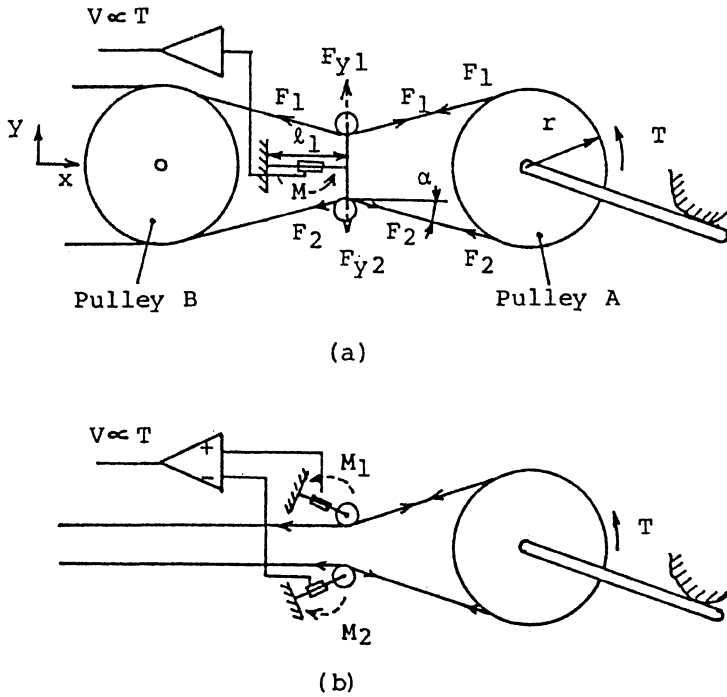


Fig. 1 Two types of torque sensors: (a) tension differential type torque sensor (b) conventional type torque sensor

This sensor is composed of two idler pulley parts and a cantilever beam part incorporating two strain gauges. When a torque is applied to pulley A, the y-directional force at the tension pulleys is expressed by

$$F_{yi} = 2F_i \sin \alpha \quad (i = 1, 2) \quad (2)$$

Then, the moment acting on the beam with strain gauges is given by

$$M = (F_1 - F_2)l_I \sin \alpha \quad (3)$$

By substituting $a = l_I \sin \alpha / r$ into the above equations, we can obtain the following expression

$$M = aT \quad (4)$$

Eq. (4) means that the strain gauge output is proportional to the torque applied around the drive pulley.

Figure 1(b) shows a popular torque sensor for tendon-pulley systems, in which two individual sensors detect tensions of the agonist and the antagonist tendons and the difference of those tensions is calculated in a differential amplifier to get the torque output. Comparing it with the proposed sensor, the following advantages of the TDT sensor can be identified:

- (1) The number of strain gauges is reduced by half.
- (2) The sensor consists of a single element and it is possible to make it smaller.
- (3) The sensor beam receives a moment only when a torque is added at the drive pulley, while tendon-tension always acts on the each sensing element in Fig. 1(b). This is effective to avoid residual stress of the sensor which may make unfavorable nonlinear characteristics.

Figure 2 shows a general view of the proposed TDT sensor. The static characteristics of this sensor are shown in Fig. 3, where the white circle plots are the results for the sensor with the pulley B as shown in Fig. 1. The black plots are obtained by removing the pulley B. The linearities between applied torque and sensor output are fairly good for both experiments. Even without pulley B, the sensor can detect the torque without a remarkable reduction of the sensitivity. Therefore, it will be possible to remove the pulley B to simplify the structure.

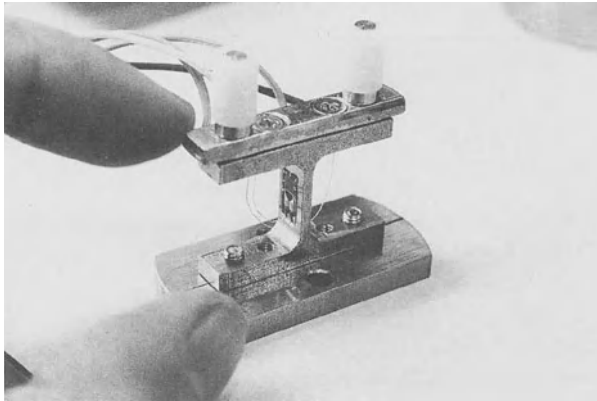


Fig. 2 A prototype TDT sensor

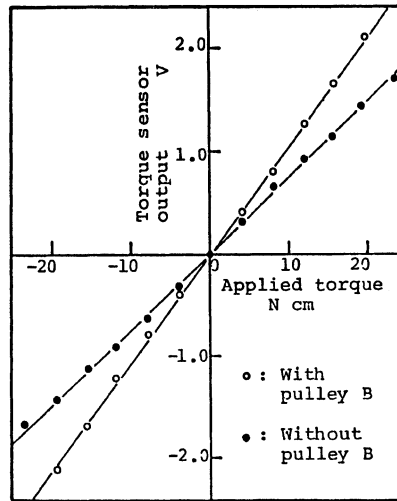


Fig. 3 Static characteristics of a TDT sensor

3. Stiffness Based Stable Grasp

3.1 Theoretical Approach

Figure 4 shows an object grasped by two fingers without friction at each finger tip. Suppose that the compliance at each finger tip can be controlled and that an object is supported by two fingers through the compliance. In the figure each finger is shown as a spring because of emphasizing the compliance effects. It is assumed that the influence of gravity is ignored for simplicity. In the non-friction model as shown in Fig. 4(a), finger tip forces can be applied only in directions normal to the surfaces. When an external force is applied to the object, making it rotate in the counterclockwise direction precisely, it is found from the geometrical structure that a restoring moment will be generated. Therefore, the grasp becomes stable for such a small rotation. Figure 4(b) is a more practical model in which friction exists at the finger tips. In this case forces applied at the finger tip are not normal to the object surface. When tangential forces less than friction forces are applied on the object surface, the finger tips will move together with the grasped object according to the applied external forces. Figure 4(b) shows the behavior of the grasping system that the counterclockwise external torque is applied to. From the observation of the behavior, the reaction forces/torques work to encourage the external torque and result in unstable grasping. As revealed by this simple example, even a grasping system which is stable in the absence of friction may become unstable if friction effects are introduced. This paper proposes an approach to make a grasping system with stable friction effects.

An n -fingered two-dimensional grasping model as shown in Fig. 5 is used for analytical discussion. Each finger is expressed by two virtual springs (k_{xi} and k_{yi}) at each finger(i) tip

----> Finger tip force in equilibrium state
 ----> Finger tip force in a small rotation θ

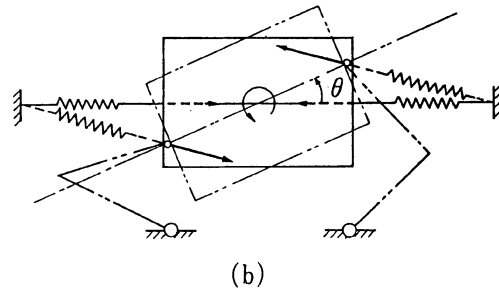
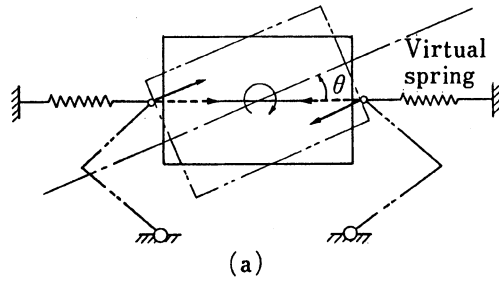


Fig. 4 Comparison in stability between the (a) non-friction grasp model and the (b) friction grasp model

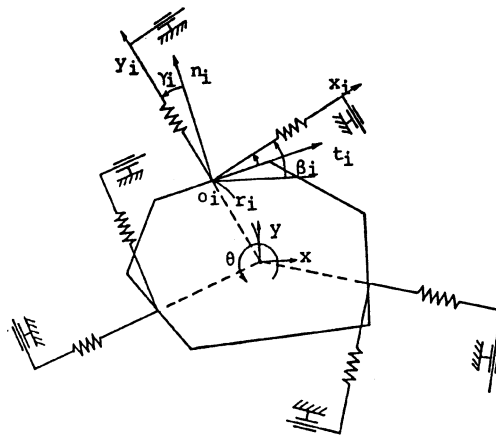


Fig. 5 Virtual stiffness based on an analysis model

because it is assumed that the compliance at each finger tip can be adjusted. A finger tip i has a coordinate system $(o_i-x_iy_i)$ where the origin o_i is at the contact point of the finger tip and an object. The k_{x_i} and k_{y_i} are stiffness parameters along x_i and y_i axes, respectively. The directions of the coordinate axes x_i , y_i will be determined from the compliance center of the object which can be put arbitrarily inside the object. The axis y_i is selected along a line which goes through both finger tips, and the compliance center and x_i is perpendicular to y_i . These precautions are worthwhile in a discussion of the grasp stability in a simple way.

Suppose that the finger tip i will be moved along the y_i axis to hold an object. When the virtual position reference along the y_i axis exists in the object, the virtual spring along the y_i axis will be deformed and will generate an internal force which corresponds to the grasping force of the object. On the other hand, the virtual spring along the x_i axis keeps natural length, if no external forces are applied and, therefore, generates no applied force to the object. However, the virtual spring along the x_i axis plays an important role in keeping a stable grasp under a small rotation of the object, because the virtual spring along the x_i axis always generates a restoring moment against a rotation of the object. These discussions are formulated in detail in the following.

The stiffness matrix K of the grasped object in Fig. 5 can be expressed in the following form,

$$K = \begin{bmatrix} k_{xx} & k_{xy} & k_{x\theta} \\ k_{yx} & k_{yy} & k_{y\theta} \\ k_{\theta x} & k_{\theta y} & k_{\theta} \end{bmatrix} \quad (5)$$

Now, for simplicity, assume a decoupling condition between rotational motion and translational motion. This condition is given by,

$$k_{x\theta} = k_{\theta x} = 0 \quad (6)$$

$$k_{y\theta} = k_{\theta y} = 0 \quad (7)$$

By exploring the positive definiteness of the stiffness matrix K , the following stability condition is obtained:

$$k_{\theta} > 0 \quad (8)$$

This condition can be also expressed by using virtual spring constants,

$$\sum (k_{x_i} r_i^2 - k_{y_i} \delta_{o_i} r_i) > 0 \quad (9)$$

Where r_i , and δ_{o_i} are the distance between the compliance center and finger tip, and the initial compression of the virtual spring along the y_i axis, respectively. From Eq.(9), it can also be understood that the virtual spring along the x_i axis plays an important role for stable

grasping. For example, suppose that in the two-fingered model of Fig.4(b), there are relations, $k_{x1}=k_{x2}=0$. In this case, $k_\theta = -\sum k_{yi}\delta_{oi}r_i < 0$ can be obtained from Eq.(9). Therefore, it is easily found that the grasping system becomes unstable. From this fact, it is confirmed that the existence of the virtual spring parameter along the x_i axis is important for stable grasping.

3.2 Experimental Verification

The derived stability condition will be verified experimentally by using the developed two-fingered robot hand with the capability of compliance control. The developed robot hand has three joints for each finger. In this research, however, the base joint is mechanically locked and only two joints for each finger are used. The TDT sensor specially designed for this robot finger is installed at a location close to each joint to measure the joint torque. Using this torque sensor, a torque feedback system is constructed at each joint. In order to provide a desired stiffness with the finger tip, each joint torque is controlled by the Active-Stiffness-Control method proposed by Salisbury[8]. When the desired stiffness matrix at the tip of finger i is K_i , the joint stiffness matrix K_{qi} is adjusted according to the following relation,

$$K_{qi} = J_i^T K_i J_i \quad (10)$$

where J_i is Jacobian matrix for i -th finger. T shows the transpose of the matrix. Using the above equation, joint torque control law can be written as the following equation,

$$T_i = K_{qi}(q_{ri} - q_i) \quad (11)$$

where T_i , q_{ri} and q_i are the joint torque vector, the reference input vector of joint angle, and the current joint angle vector, respectively, for the finger i . To manipulate the object, the desired trajectory of the object is transformed to a trajectory in the joint space of each finger i using the proper inverse kinematic equations. This joint trajectory is used for the reference input vector q_{ri} . Figure 6 shows the block diagram of the control system.

In the experiments, at the beginning two fingers grasp the object with a small offset angle against the horizontal direction. After the grasp parameters are set to the horizontal direction, the motion of the object is observed using the trajectories of LEDs installed at the finger tip. Figure 7 shows two typical experimental results, where Fig. 7(a) corresponds to stable condition ($k_\theta = 376 \text{ Nmm/rad}$), and Fig.7(b) unstable condition ($k_\theta = -114 \text{ Nmm/rad}$). The photographs have been made in a dark room keeping the camera shutter open, and the light is flashed at both initial and final states. In spite of a relatively large offset angle, Fig. 7(a) shows the way in which the object is recovering from an offset state to an equilibrium state. On the other hand, Fig. 7(b) shows the way in which the grasp is collapsing for a small offset angle.

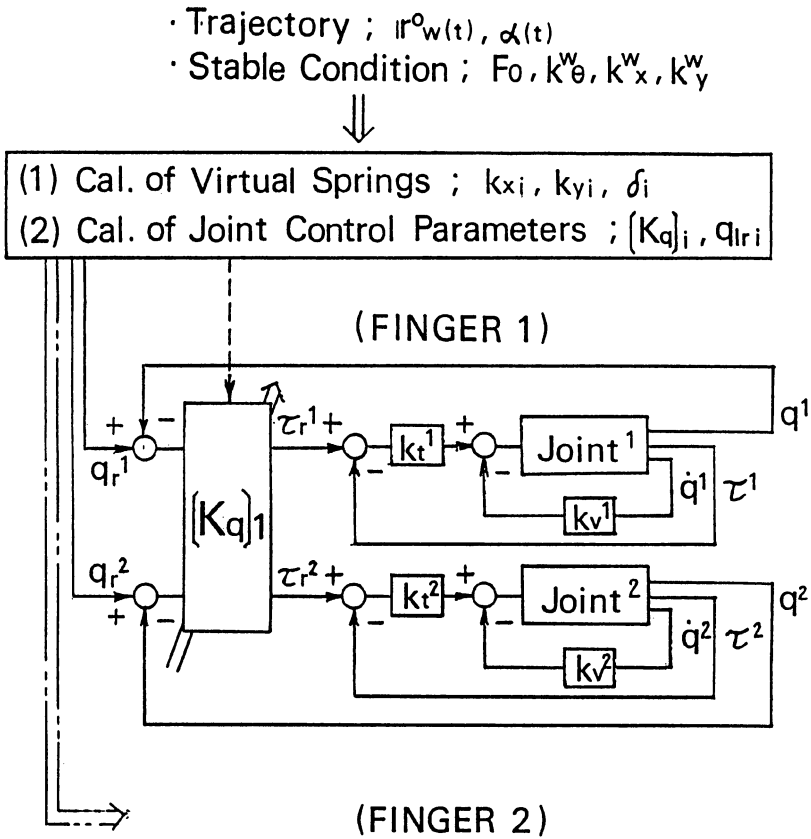


Fig. 6 Control system

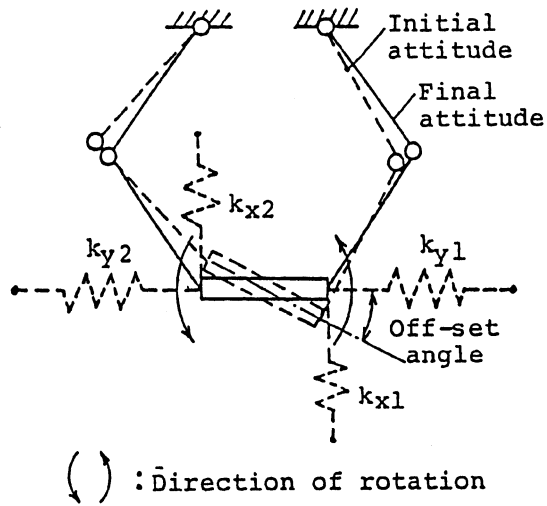
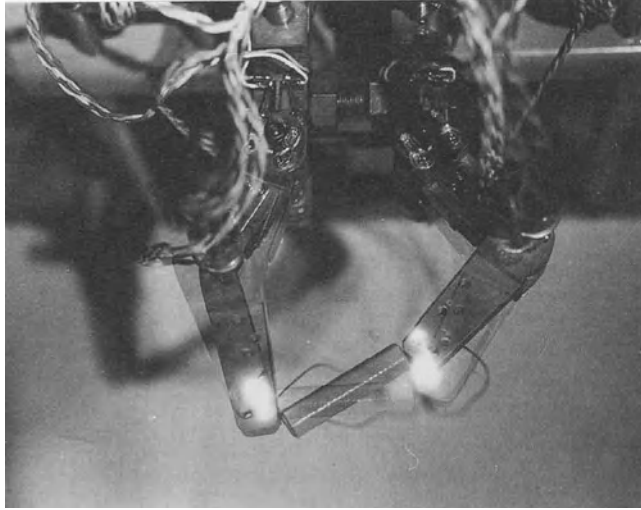


Fig. 7 Grasping experiments: (a) stable condition ($k_0=376$ Nmm/rad)

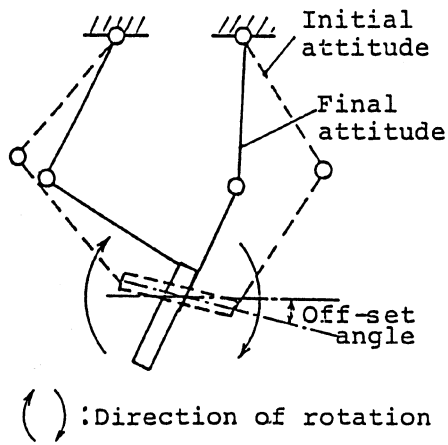
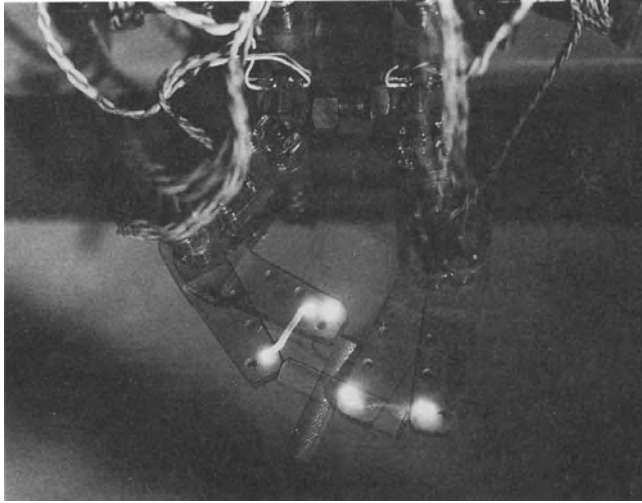


Fig. 7 Grasping experiments: (b) unstable condition ($k_\theta = -114 \text{ Nmm/rad}$)

4. Conclusions

The results of this paper are summarized as the following.

- (1) A Tension Differential type Torque sensor (TDT sensor) has been proposed suggested by the idea that the torque around a drive pulley is proportional to the tension difference, which can be measured directly without sensing individual tendon-tensions.
- (2) A stiffness model based stable grasp has been proposed, and using stiffness parameters, it was found that a stable grasping condition could be described in a simple form.
- (3) A method of manipulating an object in keeping with the stable grasping condition has been proposed.
- (4) A robotic hand system with two articulated fingers and the capability of compliance control has been developed, and using the system analytical results have been confirmed.

References

1. Baker, B. S., Fortune, S. J., Grosse, E. H.: Stable Prehension with a Multi-Fingered Hand, Proc. IEEE Int. Conf. on Robotics and Automation, St. Louis, 570-575, 1985.
2. Hanafusa, H., Asada, H.: Stable Prehension By a Robot Hand with Elastic Fingers, Proc. of 7th Int. Symp. on Industrial Robots, Tokyo, 361-368(1977).
3. Jacobsen, S. C., Iversen, E. K., Knutti, R. T.: Design of the Utah/M.I.T. Dexterous Hand, Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, 1520-1532(1986).
4. Dario, P., G. Buttazzo, G.: An Anthropomorphic Robot Finger for Investigating Artificial Tactile Perception, Int. Journal of Robotics Research, Vol. 6, No.3, 25-48(1987).
5. Kobayashi, H.: Control and Geometrical Consideration for an Articulated Robot Hand, Int. Journal of Robotics Research, Vol. 4, No.1, 3-21(1985).
6. Nguyen, V.: The Synthesis of Stable Grasps in the Plane, Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, 884-889(1986).
7. Okada, T.: Object-Handling System for Manual Industry, IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-9, No. 2, 79-89(1979).
8. Salisbury, J. K.: Active Stiffness Control of a Manipulator in Cartesian Coordinates, Proc. 19th IEEE Conf. on Decision and Control, 95-100(1980).
9. Salisbury, J. K., Craig, J. J.: Articulated Hands: Force Control and Kinematic Issues, Int. Journal of Robotics Research, Vol. 1, No. 4, 4-17(1982).
10. Tsusaka, Y., Sawasaka, N., Inoue, H.: Turning an Object between Reposing Postures, Proc. 5th Annual Meeting of the Robotics Society of Japan(in Japanese), Tukuba, 451-452(1987).

Part 3

Locomotion

MOBILE ROBOTS - THE LESSONS FROM NATURE

D. J. Todd

Department of Mechanical Engineering
University of Edinburgh
King's Buildings
Mayfield Road
Edinburgh EH9 3JL
Scotland

1. ABSTRACT

This paper records some of the ways in which biology has contributed to the technology of locomotion and discusses a number of features found in animals which have yet to be fully exploited. These subjects are illustrated with reference to the design of a legged vehicle being developed at Edinburgh. The paper goes on to list a number of additional design ideas drawn from nature which seem to have potential for future legged robots.

2. INTRODUCTION

The connection between biology and robotics is at its most striking in the phenomenon of locomotion. Locomotion is the hall-mark of the animal world, and it is when we make mobile robots that we seem to be emulating biology most closely; here, if anywhere, we ought to be able to learn from nature.

This has long been acknowledged and, especially in the case of locomotion with legs, many principles derived from the observation of legged animals have consciously or unconsciously been incorporated into the design of legged robots.

This paper records some of the ways in which biology has contributed to the technology of locomotion and discusses a number of features found in animals which have yet to be fully exploited.

It proceeds largely by discussing the ways in which biological observations have influenced the design of a particular robot, and by listing other animal mechanisms which have not yet been developed to any great extent, although tentative trials have been made of some of them.

Since most of my work is on legged locomotion, my observations of nature have been concentrated on how animals walk and run, and these observations are relevant almost exclusively to legged vehicles. However, one cannot help noticing some behavioural aspects of animal life, which would seem to be relevant to all kinds of mobile robot, whatever their locomotion method.

Particularly interesting is the way in which animal behaviour is usually appropriate: animals are much less likely than machines to walk off cliffs, get stuck in corners or load a

damaged limb to destruction. The final section of the paper suggests that some current research, together with aspects still to be investigated, can be regarded as constituting a discipline of *ethorobotics*, and that an ethorobotic perspective is a good way of handling complexity.

3. THE INFLUENCE OF NATURE ON THE DIRECTION OF LEGGED ROBOT RESEARCH

In the early days of research on artificial legged locomotion, there were two main themes: first, prosthetic and orthotic aids to human walking, together with man-amplifiers or powered exoskeletons; and, second, legged vehicles for rough-terrain transport.

The first theme is so closely bound up with human locomotion that it hardly makes sense to speak of lessons from nature: the whole enterprise is one of assisting nature.

In the second, however, we see a direct attempt to imitate nature in machines. The idea that intermittent support by the cyclic motions of a set of levers is a feasible and appropriate method of locomotion is itself the first and most crucial, if largely unconscious, observation from nature. It was observations of the high mobility of legged animals which led to the view that legged machines could have advantages over wheeled ones; the main advantages being the ability to travel on discontinuous surfaces, and the superior efficiency of legs compared with wheels on soft ground.

One of the most decisive influences nature has had is in the choice of leg number and geometry. There is no theoretical upper limit to the number of legs, or on where they should be placed, or on how many joints they should have. But since evolution selects only viable designs (although it may miss some, as it missed the wheel), nature provides us with examples which we know can be effective. We are all familiar with examples of impressive legged locomotion configurations: the horse, the beetle, the spider, ourselves. The direction taken by the pioneers of research on legged robots was largely governed by these examples.

Once a mechanical configuration has been chosen, the next issues are the interlinked ones of balance and gait. Here again, the possibilities are infinite but nature has given us examples of effective gaits, such as the alternating tripod gait of hexapods, and the trot and gallop in quadrupeds. Of course, once these have been identified in nature it is possible to formulate a systematic theory, in some cases finding that by changing one or two parameters it is possible to generate a series of apparently different gaits from a single formula.

These influences are readily apparent in the design of the first walking robots. Even machines radically different from anything in nature, such as Raibert's monopod (Raibert 1986), have been largely inspired by a desire to understand and eventually imitate natural locomotion. In this case the apparent departure from natural models is actually a device for studying natural behaviours such as hopping and running in a simplified form.

Rather than attempt an exhaustive list of the influences of biology on machine design, I shall illustrate the theme by describing the way in which a specific machine incorporated some lessons from nature, then go on to deal with some under-exploited mechanisms.

4. INFLUENCES ON THE HYDRAULIC TEST VEHICLE

This machine, which has been under development in the Department of Mechanical Engineering at Edinburgh University for about three years, was intended as a general-purpose test rig to allow us to study various issues on a scale relevant to the design of large vehicles, while at the same time keeping costs to a minimum. It is not particularly original in concept: the leg design uses a pantograph derived from the Ohio State University Adaptive Suspension Vehicle (Song and Waldron 1989), and its speed and agility are likely to remain poorer than that of many existing machines. However, since it uses relatively cheap components and simple control methods, it may pave the way for an extension of the areas in which legged robots will be economic.

The general arrangement of the machine is shown in Figure 1. The design was influenced by nature in several respects. First, the structure is like a vertebrate, based on a narrow spine rather than a space frame. This structure was chosen because of the extra agility conferred on an animal by bending the spine. We envisaged that one day it might be desirable to introduce joints into the body structure, and this is easier with a spine than with a framework.

A second strong influence was in the choice of gait. The machine has been operated with four legs and with six. As a quadruped, lateral stability is a significant problem in a narrow and not very rigid machine. Animal gaits were extensively studied in an attempt to find the most suitable. The ideal gait for such a machine, in terms of speed, is the trot, but this is difficult to stabilize at low speeds. What is needed is a smooth transition from a crawl (in which only one leg at a time is lifted) to a trot. The work reported in Stuart et al (1973) shows this is possible, and occurs in cats. A gait with these characteristics is shown in Figure 2.

In hexapod mode the machine's main gait is the alternating tripod, well known in insects; see, for example, Graham (1985), or, better still, go out and watch a beetle.

Another way in which observations from nature influenced design was in the leg disposition in hexapod mode. The question is, in going from four legs to six, where should the two extra legs be located? This can be decided by noting that arthropods with many legs face the problem of interference between adjacent legs. One solution, common for leg numbers between six and perhaps twenty (but not seen with the very large leg numbers found in millipedes), is to offset the legs so they can overlap each other. (See Figure 3; also Manton(1977)). In the case of hexapods this has the additional advantage that when standing on a tripod of legs the load distribution is made more even, so the more distal sections of the middle legs need be no stronger than those of the end ones. Further, lateral stability is improved because the base of support is wider (Figure 4).

Other design features based on nature have been considered for this vehicle: for example, primitive experiments were made on the use of a 'head' and 'tail' as balancing aids, and of wide feet with locking ankle joints to improve lateral stability. These principles are not currently implemented, and are described in the next section.

5. FURTHER LESSONS FROM NATURE

This section lists some biological mechanisms which have yet to be exploited by robot designers to any extent.

Balancing aids

Machines such as Ohio State University's ASV can to some extent adapt to sloping ground by adjusting differential leg height or average leg angle, but this may have undesirable consequences such as limiting step length. An alternative or supplementary method of balancing on slopes is to move the centre of mass of the machine by extending, swivelling or bending extra appendages. The problems of balance in the sagittal and frontal planes are similar, but they differ in that long members sticking out sideways are undesirable as they would prevent passage through narrow gaps. Figure 5 shows how a weight-carrying flexible appendage at each end of the vehicle could be used to move the centre of mass both laterally and longitudinally. Such balancing devices might also be useful for obstacle crossing, by allowing normally unstable postures (Figure 6).

Substrate reaction

A flexible or jointed neck and tail could also be used to push against the ground. If a machine were in danger of overbalancing it could brace an appendage against the ground while seeking a new foothold; if the appendage were prehensile or equipped with a gripper (as the arms might be on some robots) it could perhaps be used to pull the robot up a slope or out of a ditch. A more dramatic use would be returning a robot to its feet if it fell over completely (Figure 7).

Crab leg arrangement

A six- or eight-legged machine with a row of three or four legs at each end, instead of a row down each side as is usual, would be relatively narrow and able to pass through confined spaces. The Odex I robot made by Odetics Inc. can adopt this configuration for going through doorways. This arrangement is similar to that of a crab when it walks sideways. Crabs presumably walk sideways for the same reason; given their body shape, walking sideways lets them pass through narrow apertures.

Spine flexing

All mobile robots so far have had a rigid body (apart from one or two articulated wheeled vehicles proposed for planetary exploration). All vertebrates, however, can bend the body to some degree. An obvious advantage is the ability to bend round obstacles, snakes being the ultimate example of this, but there are other advantages. Bending the body can aid in getting up from a lying position, it can help in reaching with the head or arms, and it can increase the speed of locomotion. The supreme example of this last use is the cheetah. On each stride the spine, by bending, effectively changes its length, adding to the stride. In addition, its bending effectively introduces another joint at the hip and shoulder, allowing the leg to swing through a greater angle, and again increasing the stride length. Something similar happens in salamanders, but with the spine flexing in a horizontal plane (Hildebrand 1974).

Folding Feet

For robots which walk slowly enough for there to be significant phases relying on static stability, the size of the feet, both laterally and lengthwise, can be important. This is particularly so for bipeds.

Bipedal animals, including man, tend to have relatively large feet. In particular, birds such as ducks clearly use the width of the foot with outspread toes as a means to stability. The problem is that such wide feet tend to interfere with each other during walking.

The solution some birds adopt is to fold up the foot when it is raised on every step so it is compact as it passes the supporting leg; the toes are then spread again as the foot is lowered. This simple mechanism would seem quite suitable for some legged robots (Figure 8).

Conformal Locomotion

Observation of animals in confined spaces suggests that one of the reasons they need so much less space than vehicles such as automated guided vehicles (AGVs) is that they can operate in contact with obstructions, and even gain support and propulsion from them. The ultimate example is the snake, which propels itself entirely by reacting its body against the environment.

A *conformal vehicle* is one which conforms to the geometry of the fixed objects it encounters, by sensing them and altering its path and often its own shape. Such a shape change may vary, from simply altering the track width of a wheeled vehicle, to the sinuous bending of a snake. (At least one robot has been made using the snake principle: see Hirose and Umetani (1977)). An intermediate case is that of a legged vehicle whose legs can be folded or extended to get them past obstacles.

A conformal vehicle could squeeze through a narrow gap; it could lean on walls if this helped (a method not unknown in humans in conditions of reduced sensorimotor ability); and make efficient use of the space in tight corners.

Conformability rests on a combination of principles:

- The use of support from walls;
- The ability to brush past or along a surface using a 'fur' of touch or proximity sensors;
- The ability to back-track a sequence of actions when blocked and try again;
- A reflex-like folding of the legs when passing an obstacle.

There would seem to be a large field of research open here.

Soil Testing

When one walks on a beach or an icy pavement one tests certain properties of the substrate on every step, although usually without being very conscious of it. These tests, mainly of sinkage and friction, enable one to modify one's gait and route so as to minimise effort or reduce the danger of bogging down or slipping.

It would be useful for a walking robot to similarly sense the properties of the soil on each step. The most obvious properties are sinkage and friction.

Relationships between depth of sinkage and pressure have been derived for various soil models in studies of building foundations and the mobility of wheeled and tracked vehicles. Using such a relationship, with experimentally determined parameters, it would be possible to take the sinkage with a known foot loading and predict the sinkage under other loadings; or a cruder analysis might suffice. In any case, some means of measuring foot sinkage is needed.

We have experimented with a foot-mounted penetrometer, but a better method is to fit each foot with pads which rest lightly on the soil and provide a reference for measuring depth (and rate) of sinkage on each step (see Figure 9).

Finding how slippery a surface is can be done in several ways. A foot could be equipped with a probe to be dragged a short way across the surface, or with a small powered wheel, to directly measure the frictional forces at a certain load. Alternatively, the forces developed as one leg moved relative to another could be measured. Thirdly, for dynamic gaits the robot could sense the patterns of force, acceleration and displacement as each foot touched down: sudden horizontal movements would indicate slipping.

6. ETHOROBOTICS

Much research on mobile robots has consciously or, more often unconsciously, drawn on ideas arising from a study of animal behaviour. Usually these ideas have not been taken from the ethological literature but represent a rather casual view of the way animals behave. The term 'ethorobotics' is meant to denote the explicit incorporation of ethological observations into robot control systems.

What specific contributions might the ethorobotic perspective make? It might begin with the issue of autonomy, survival and appropriate behaviour. A comparison of different animals leads to the (rather obvious) conclusion that autonomy is not the same thing as intelligence. While most research in artificial intelligence (AI) has been directed towards intelligence, a mobile robot also needs a high degree of autonomy. What is striking about animals is that even simple animals such as woodlice, with very little intelligence in the sense of reasoning ability, and with relatively simple senses, are highly autonomous: they need nobody to initialize their navigation systems, or turn them on, or get them out of corners. One of the aims of the ethological approach to robotics should be to refine the concept of autonomy and to devise ways of conferring it on control structures. (For example, how important is it for an agent to be able to set its own goals?)

Autonomy is also bound up the concept of appropriate behaviour: the ability to adjust to circumstances even if unfamiliar or unintelligible. A robot should never be found stuck in a corner with its wheels grinding away because a sensor has failed. A corollary to this approach is that a robot should know when it is being interfered with, for example by being relocated, or turned over, or damaged. It may be unable to continue as normal, but at least it should be able to function in some organized way, if only by shutting itself down and waiting for rescue.

Some research in robotics, for example that of Brooks (1986), does address these issues.

A second example of the ethorobotic approach is the use of simple behavioural mechanisms such as taxes and tropisms for getting about in a disordered environment. Many invertebrates use such ways of finding food or shelter or a mate. These methods rely on detecting the intensity or gradient or direction of some environmental property, or on searching for an easily recognized marker. In a *taxis* an animal heads towards or away from a stimulus, either comparing the signal from two sensors or by using body movement to scan with a single sensor. In a *kinesis* rate of turning, or speed of locomotion, is governed by the intensity of a stimulus. A robot might search for its home base or for some target using such

methods; the target might be an unexploded mine, a gas leak, a weed or pest, a fragment of a crashed aircraft... Sensing methods could include magnetic and chemical sensors as well as touch and vision.

A related idea is the sensing of multiple properties of the environment for navigation. Animals without imaging abilities must identify places mainly by sensing a highly specific property such as smell, or a combination of properties. Some properties which could be sensed by a robot are:

- Soil temperature, colour, hardness, texture, humidity
- Air temperature, pressure and humidity
- Wind speed and direction
- Ground slope and roughness
- Ambient light level, colour, polarization
- Ambient sound level and spectrum
- Magnetic field strength and direction
- Range to nearby objects
- Smell (chemical concentration in air or soil)
- Vegetation type.

In a reasonably varied environment each place of interest may well have a unique signature if a vector of all these measurements is computed. Such signatures, together with dead reckoning, could provide the basis for a navigation system within a robot's habitual territory of operation.

A further aspect of this concept is the *marking* by an animal of places it wishes to identify, by scent or other means. A robot might plant beacons (acoustic, optical, radio or chemical) to which it could subsequently refer.

Other subjects, such as the scheduling of behaviour, on which there is already much research in AI, could also be regarded as within the domain of ethorobotics; yet other abilities, such as the use of imitation in acquiring skills, have not even begun to be explored for robotics.

7. BIBLIOGRAPHY

There is a substantial zoological literature relevant to robotics, and an adequate review would need a separate paper. This section just lists some sources the author has found particularly useful.

The mechanics and energetics of vertebrate locomotion have been extensively discussed by R. McNeill Alexander (whose publications are too numerous to list), T. A. McMahon and M. Hildebrand (see references). Some extended treatments of locomotion in invertebrates are to be found in Manton(1977), Graham (1985), Stein et al. (1973) and Herreid and Fournier. A glance through the references in these works will reveal the journals where continuing research results are published.

There is also a massive literature on ethology. Among the works the author has found helpful are Bateson and Hinde (1976), Hinde (1970) and Camhi (1984).

For the reader with an interest in legged robots there are now several books on the subject as well as the regular journal and conference publications. Four books of general interest are Raibert (1986), Song and Waldron (1989), Sutherland (1983), and Todd (1985).

References

- Alexander, R.McN. *Animal Mechanics*. Blackwell Scientific 1983
- Alexander, R.McN. *Elastic Mechanisms in Animal Movement*. Cambridge University Press 1988
- Bateson, P.P.G.; Hinde, R.A. *Growing Points in Ethology*. Cambridge University Press 1976
- Brooks, R.A. A robust layered control system for a mobile robot. *IEEE J. Robotics and Automation*, RA-2(1), 14–23 (March 1986)
- Camhi, J.M. *Neuroethology*. Sinauer Associates Inc. 1984
- Graham, D. Pattern and control of walking in insects. *Advances in Insect Physiology*, 18, 31–140 (1985)
- Herreid, C.F.; Fournier, C.R. (eds.) *Locomotion and Energetics in Arthropods*. Plenum Press
- Hildebrand, M. Motions of the running cheetah and horse. *J. Mammalogy*, 40(4), 481–495 (1959)
- Hildebrand, M. *Analysis of Vertebrate Structure*. John Wiley and Sons 1974
- Hildebrand, M.; Bramble, D.M.; Liem, K.F.; Wake, D.B. (eds.) *Functional Vertebrate Morphology*. Belknap Harvard 1985
- Hinde, R.A. *Animal Behaviour*. 2nd edition. McGraw-Hill 1970
- Hirose, S.; Umetani, Y. Kinematic control of an active cord mechanism with tactile sensors. In: 2nd CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators, pp. 241–252. Elsevier 1977
- Manton, S.M. *The Arthropoda*. Clarendon Press 1977
- McMahon, T.A. Mechanics of locomotion. *IJRR*, 3(2), 4–28 (1984)
- Raibert, M.H. *Legged Robots that Balance*. MIT Press 1986
- Song, S.-M.; Waldron, K.J.; *Machines that Walk*. MIT Press 1989
- Stein, R.B.; Pearson, K.G.; Smith, R.S.; Redford, J.B. (eds.) *Control of Posture and Locomotion*. Plenum Press 1973
- Stuart, D.G.; Withey, T.P.; Wetzell, M.C.; Goslow, G.E. (Jr.) Time constraints for inter-limb co-ordination in the cat during unrestrained locomotion. In: Stein et al. (eds.), pp. 537–560
- Sutherland, I.E. *A Walking Robot*. The Marcian Chronicles Inc. 1983
- Todd, D.J. *Walking Machines: An Introduction to Legged Robots*. Kogan Page 1985

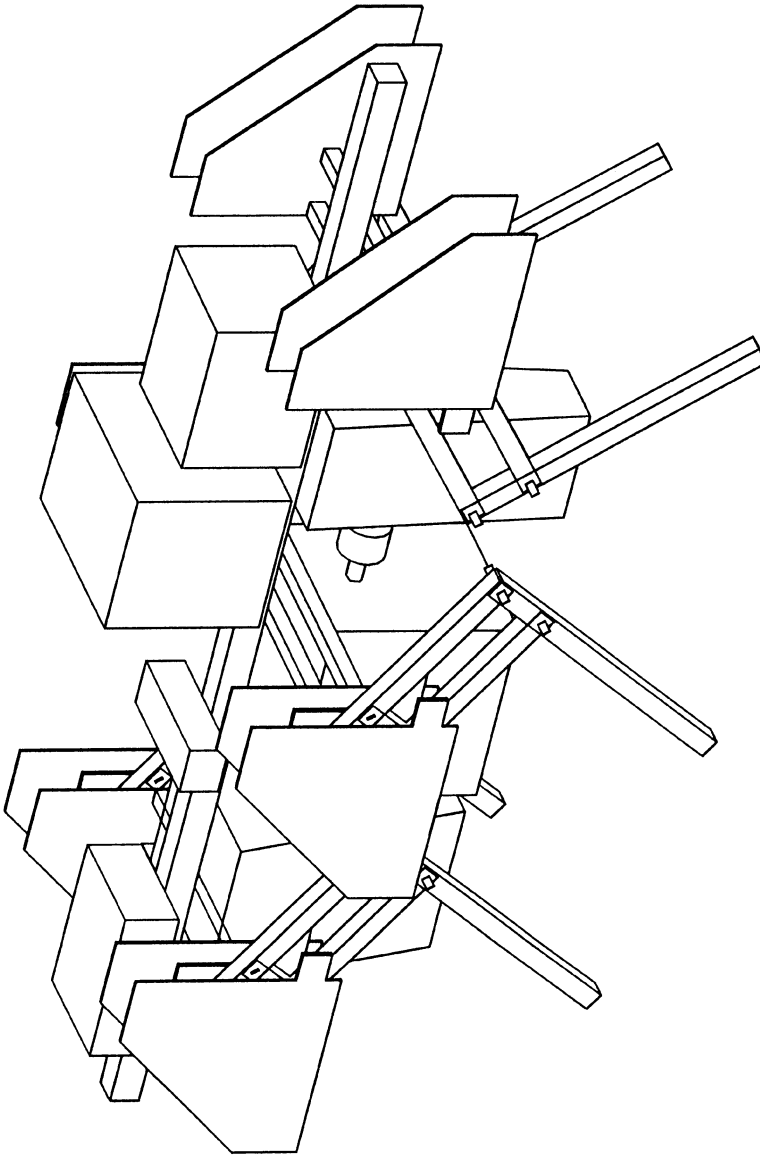
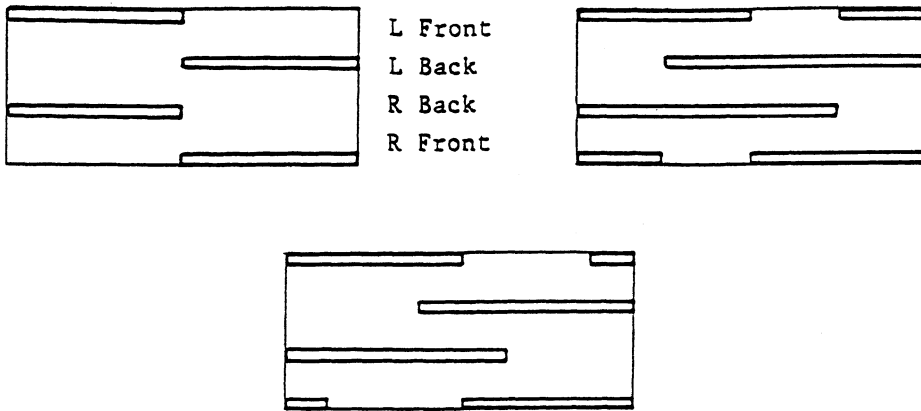


Figure 1: Hydraulic test vehicle



Three stages of a general-purpose gait. The horizontal axis represents time within one stepping cycle; the length of a bar represents the fraction of the cycle for which that foot is on the ground. The three cases shown are:

- a. Duty factor $1/2$. The trot.
- b. Duty factor $3/4$. The quadruped crawl.
- c. Duty factor $5/8$. An intermediate gait: periods of 2- and 3-leg support alternate.

Figure 2.

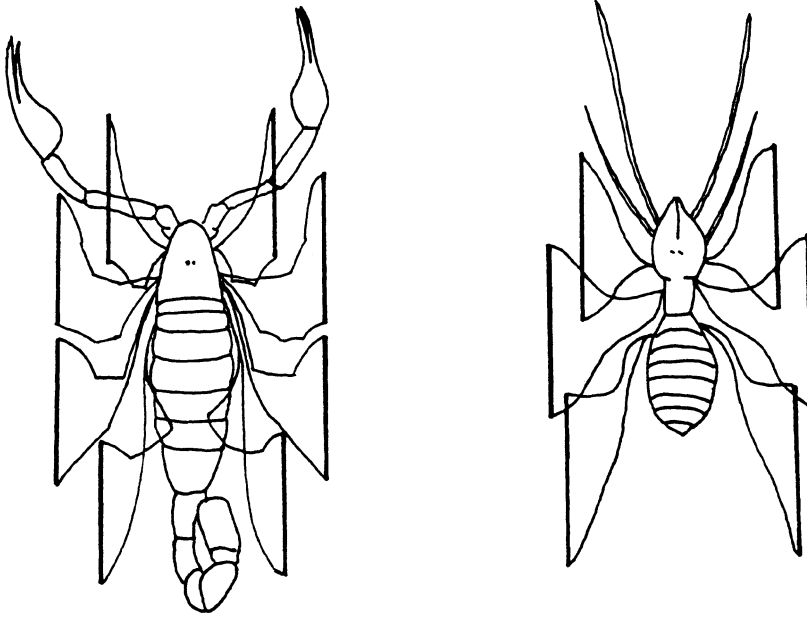


Figure 3: Eight-legged and six-legged arthropods in which the middle legs overlap the end ones

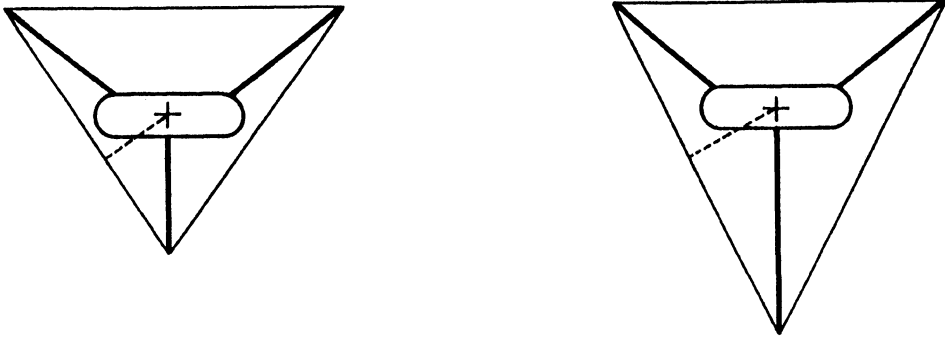


Figure 4: The effect of leg length on lateral stability. The broken line is the distance from the centre of gravity to the nearest edge of the base of support. Its length is a measure of stability. (Only the three supporting legs are shown.)

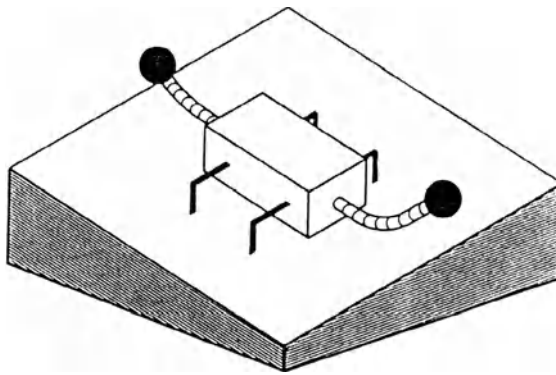
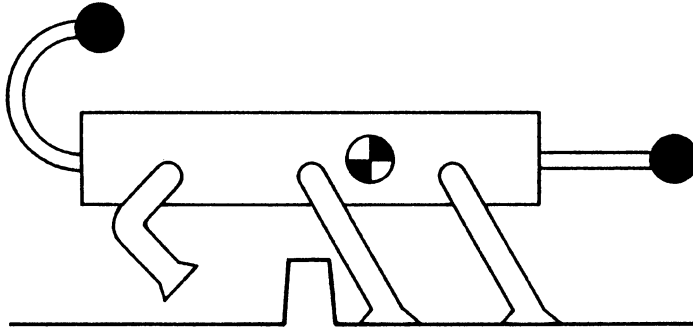


Figure 5: The use of a flexible neck and tail for both lateral and longitudinal stability on slopes



*The use of balancing aids in crossing an obstacle.
The head and tail masses move the centre of mass of the machine back while the front end is unsupported; then during the second half of the manoeuvre they swing forward to balance the unsupported rear of the machine.*

Figure 6.

end view

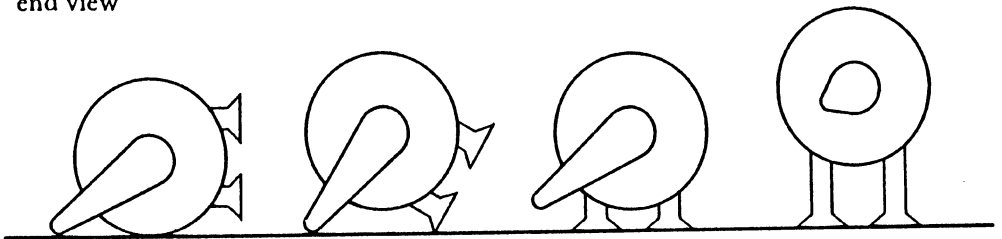


Figure 7: The use of neck- and tail-like appendages to restore an overturned robot to its feet

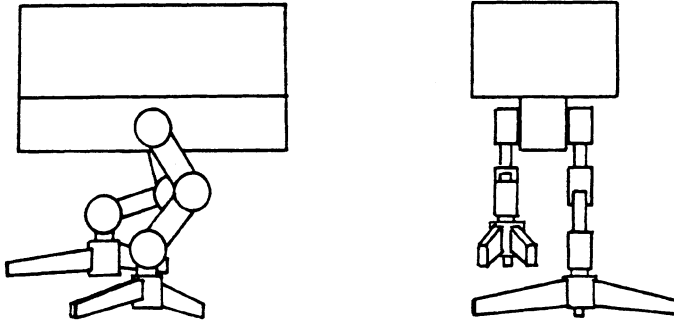


Figure 8: A wide foot which can fold to pass the other leg

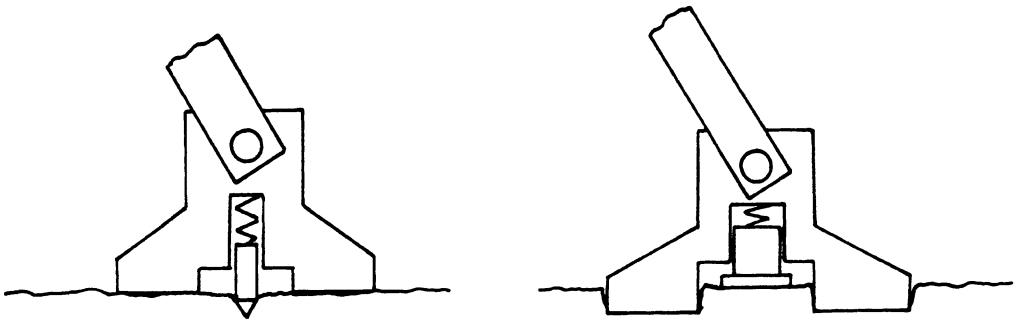


Figure 9: Two ways of measuring foot sinkage: left, a spring-loaded penetration probe attached to the foot; right, a reference plate which rests lightly on the soil so the sinkage of the main part of the foot can be measured.

Quadruped Walking Machine - Creation of the Model of Motion

A.Morecki and T.Zielinska
Institute of Aircraft Engineering and
Applied Mechanics, Warsaw University
of Technology, Al.Niepodleglosci 222,
00-665 Warszawa, Poland

Introduction

The Robotics and Technical Biomechanics Group at the Warsaw University of technology has been working on a quadruped walking machine for the last few years. The construction of the machine's legs imitates the structure of digitigrade mammals limb (e.g. horse, rabbit - Fig.1, Fig.2, Fig.3).

The machine is of static walker type [2,3], which is to say that in every instant the projection of its centre of gravity onto the ground is inside the support polygon. This way of moving is possible under an assumption that the machine moves slowly (the speed of motion does not exceed 5kmph) and the rate of change of velocity is not rapid. such a motion is easy to execute, such that most of the currently constructed machines are of the static walker type.

After the structure of the machine's legs has been assumed, the next task was to find the gait (a fixed sequence of leg transfers) that would assure the static stability [1,4] of the device.

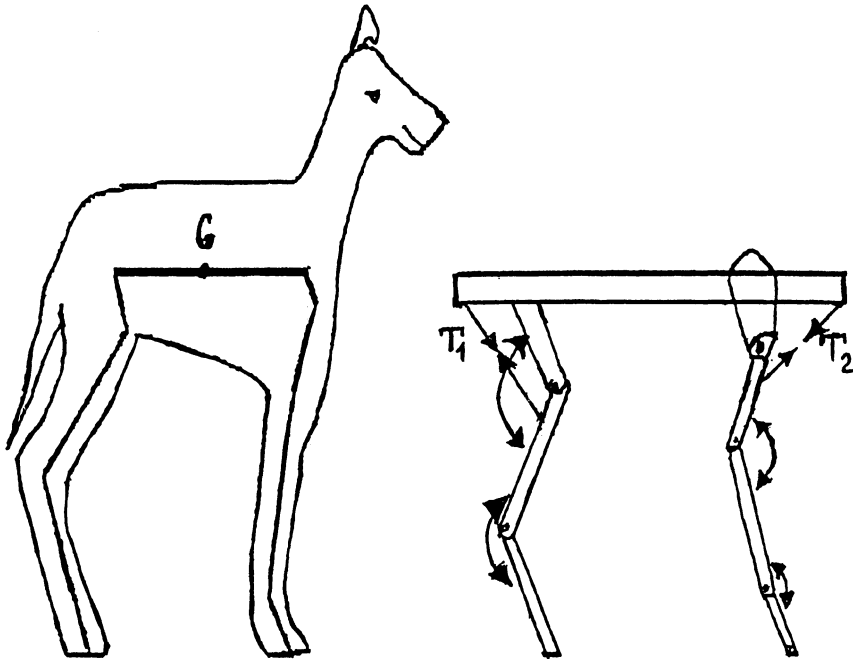


Fig. 1. Structure of the animal's legs
 G - centre of gravity of the body,
 T₁, T₂ - major muscles driving the leg.

1. The Possible Gaits

The number of potentially possible gaits is a greater as the number of legs of the walking machine grows. A.B.Bessonov and N.N.Umnov supplied the following formula describing the number of gaits as a function of the number k of legs [4]:

$$N = \sum_{i=1}^k [i^k - C_i^1 (i-1)^k + C_i^2 (i-2)^k + \dots + (-1)^{i-1} C_i^{i-1} 1^k] / i \quad (1)$$

where:

C_i^j denotes the number of combinations $\binom{i}{j}$

The number N , computed in such a way,

describes all possible gaits, even such where all the legs are above the ground (jump).

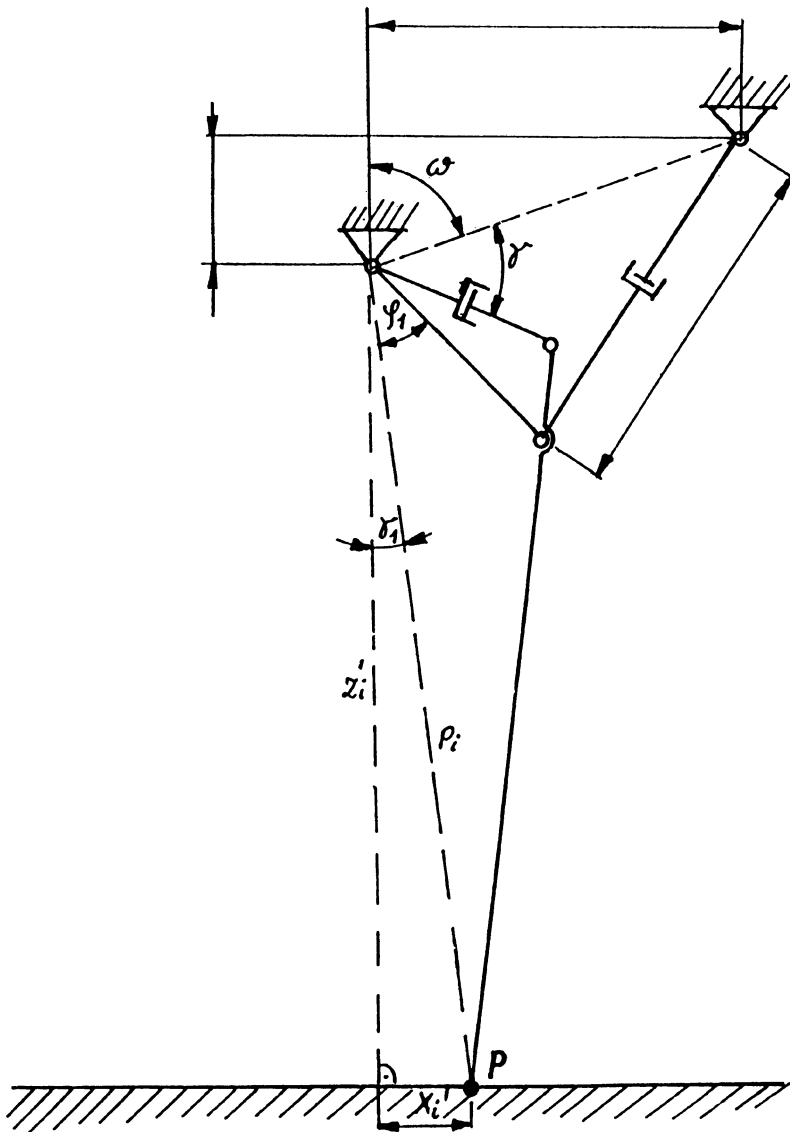


Fig. 2. Construction of the foreleg.

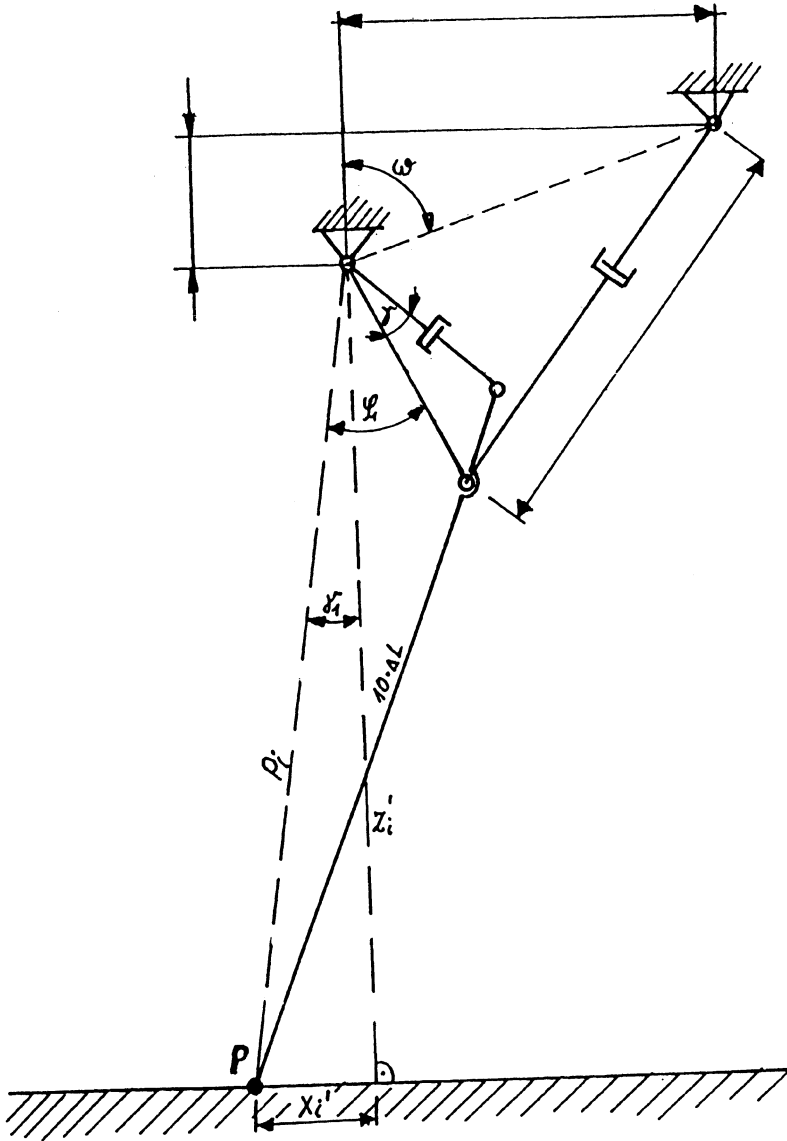


Fig. 3. Construction of the hind leg.

The number of gaits in which less than three legs are supported is equal to [4]:

$$\begin{aligned}
N^i = & (k!/(1!(k-1)!)) + k!/((k-1)!1!)+ \\
& + k!/(2!(k-2)! + k!/((k-1)!2!))+ \\
& - (k!/(1!1!(k-2)! + k!/(1!(k-2)!1!)) + \\
& + k!/((k-2)!1!1!))
\end{aligned} \tag{2}$$

The difference:

$$N^{st} = N - N^i \tag{3}$$

is equal to the number of gaits in which more than two legs are supported [4]. Among these gaits statically stable gaits should be searched for. For a four-legged machine:

$$N = 26, N^i = 20, N^{st} = 26 - 20 = 6.$$

The N^{st} gait of a quadruped walking machine are gaits with only three of four-support phases. Among these gaits a gait recommended for the designed machine was looked for.

2. The Choice of Gait

The gait parameters are as follows:

- relative time of leg-ground contact: *duty factor* - constant for each leg during the execution of the given sequence of leg transfers (during the gait),
- relative difference in time between support instants: *relative phase* - constant during each gait for the pairs of legs placed on each side of the body.

The relative values are computed in relation to the *gait period*

- the time of the execution of the sequence of leg transfers. A formula describing the value of the duty factor for the gaits which are formed by consecutive three-support and four-support phases (crawl) was elaborated:

$$r_o = 3/4 + 3M_s/(2K + 4M_s) \tag{4}$$

where K is the length of the virtual step [1] and M_s is the static stability margin

- [4] - Fig. 4.

Distinguishing the instants in which the static stability margin is minimum, coefficients r_1 and r_2 were defined. The static stability margin has a minimum value when the foreleg-end (foot) shifts in relation to the body by the distance $r_1 K$ measured from the position of the last three-support phase. Similarly the distance $(1-r_2)K$ is determined for the hind leg - Fig. 5. The foreleg and the hind leg placed on the diagonal of the body are considered then.

The values of coefficients r_1 and r_2 are different for different statically stable gaits.

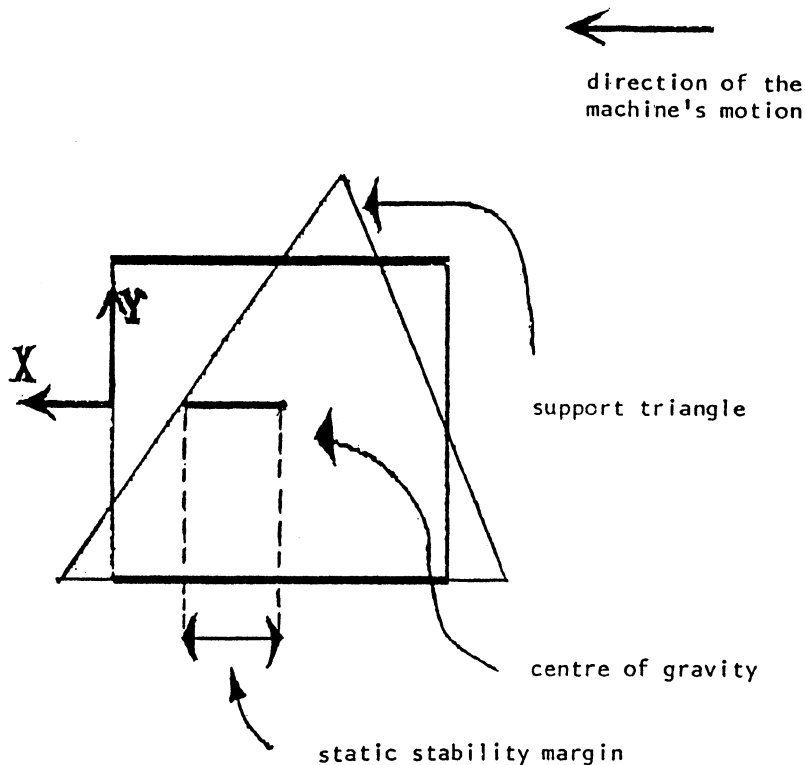


Fig. 4. Static stability margin.

Let us consider the possible gaits (possible sequences of leg transfers):

- sequence 0: left foreleg, right hind leg, right foreleg, left hind leg,
- sequence A: left foreleg, left hind leg, right hind leg, right foreleg,
- sequence B: left foreleg, right foreleg, left hind leg, right hind leg,
- sequence D: left foreleg, left hind leg, right foreleg, right hind leg,
- sequence E: left foreleg, right hind leg, left hind leg, right foreleg.

It is easily noticed that sequences D and E do not maintain the static stability (the succeeding support triangles do not have common points). For the order sequences the duty factor is represented by formula (4).

The coefficients r_1 and r_2 differ for different gaits:

- sequence 0:

$$r_1 = (2K - 2M_S) / 3K, \quad 0.5 < r_1 < 0.6 \quad (5)$$

$$r_2 = r_1 + 2M_S / K \quad (6)$$

- other sequences:

$$r_1 = (K - 4M_S) / 3K, \quad 0.0 < r_1 < 1/3 \quad (5')$$

The following relationships take place:

- sequence A: $r_2 = 0.0$
- sequence B: $r_2 = 1.0 - r_1$ (7)
- sequence C: $r_2 = 0.0$

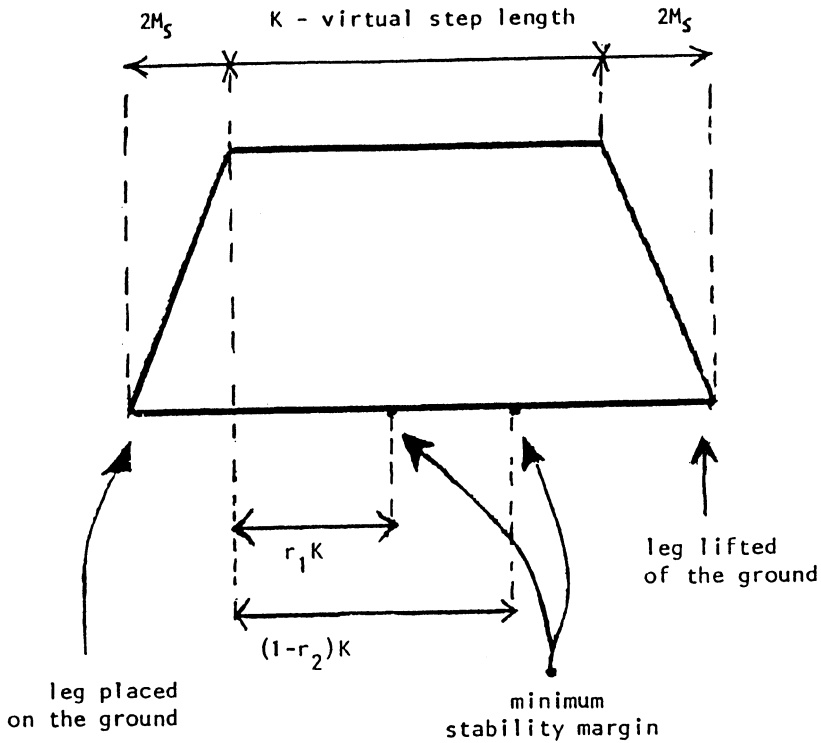


Fig. 5. Leg-end trajectory relative to the body.

Taking into consideration the constraints imposed on the leg motion (maximum step length) and finding the relationship between the static stability margin and the projection of the centre of gravity onto the ground, we obtain:

$$(S_1 + (1 - r_1) K) \leq x_C' \leq (S_2 - r_1 K) \quad (8)$$

$$(S_3 + r_2 K) \leq x_C' \leq (S_4 - (1 - r_2) K) \quad (9)$$

where:

$x_C' \leq 0.0$ - coordinate of the projection of the machine's centre of gravity onto the ground [4],

$$x_C' = x_C - H \tan(a),$$

x_c - coordinate of the centre of gravity,

H - height of the machine [4],

a - slope of the ground.

S_1, S_2, S_3, S_4 - coefficients being the functions of geometric parameters of the machine and the values of quantities determining the motion ranges of the legs.

$$S_1 = -D - S_p, \quad S_2 = D - S_p,$$

$$S_3 = -D - S_t, \quad S_4 = D - S_t$$

S_p, S_t are the sums of the geometric parameters of the machine, D is the parameter determining the leg range of motion.

The diagrams presenting the ranges of variability of x_c for the permissible values of the coefficient r_1 for the gaits 0, A, B, C are shown in Fig. 6.

Fig. 7 shows the analogous ranges for the relationship between x_c and r_2 (inequality (9)). The range of variability of x_c in the gaits A, B and C is displaced towards the positive values in relation to the range shown in Fig. 7.

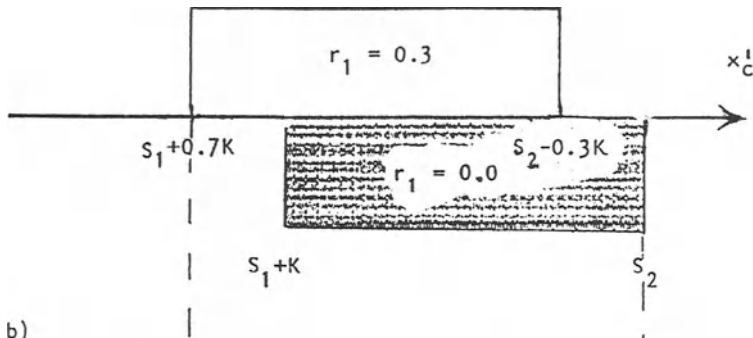
The displacement of x_c towards the positive values corresponds to a more steep slope of terrain while the machine is descending ($a < 0.0$).

From this and due to the constraints (8) results that the gaits A, B and C can be used when the slope of terrain is smaller than the slope used with gait 0. The analysis of relationship (9) (Fig. 7 a), b)) leads us to the conclusion that, due to the displacement of x_c towards the negative values, the gaits A and C (Fig. 7) should be used for the slopes of the terrain larger than the slope appropriate for the gait 0. The gaits A and C restrict the motion capabilities (in respect to the permissible terrain slopes) in comparison with the gait 0.

We have asserted that due to condition (8) the gait B should be used for smaller slopes than when using gait 0. In the case of different machines the gait 0 should be used when the machine's centre of gravity is shifted to the back of the body and otherwise gait B should be used (the length and width of the machines have to be the same).

In respect to condition (9) the gaits B and 0 are comparable.

a)



b)

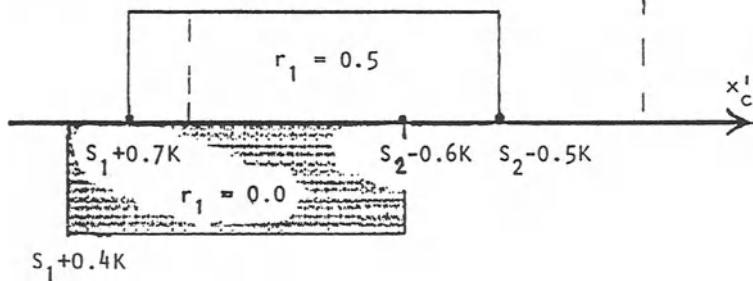


Fig. 6. The range of variability of x'_c as a function of r_1 : a) gaits A, B, C, b) gait 0.

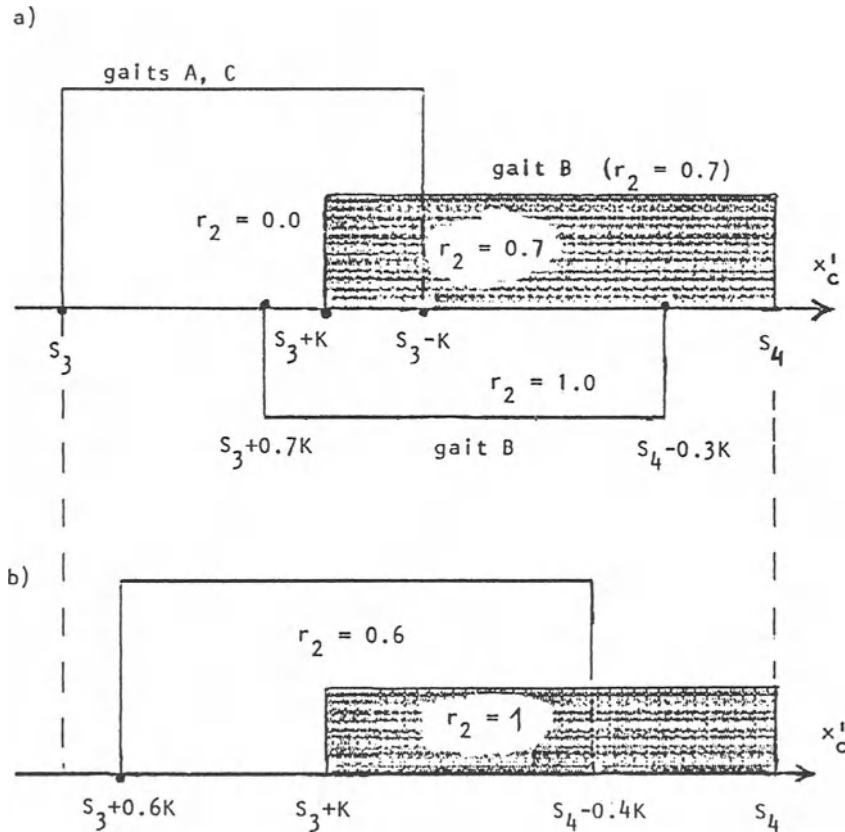


Fig. 7. The range of variability of x'_C as a function of r_2 : a) gaits A, B, C, b) gait 0.

As the variability of the coordinate of projection of the centre of gravity x'_C is larger for gait 0, this sequence was assumed to be the gait for the designed machine. This is the only statically stable gait observed in the animal (quadruped) world. This kind of gait is called the *quadruped crawl*.

In this way, when moving very slowly, moves e.g. horse, cow, turtle.

3. The Choice of the Gait Model

After the kind of gait (the sequence of legs transfers) had been chosen a method of its description was elaborated [5]. This description gives the relationship between the position of each leg and the actual instant of motion. The description takes into account that the legs of machine can move only in one plane.

The speed of motion is equal to:

$$v = (4K + 4M_s) / 3T \quad (10)$$

where T is the *gait period* (the time of execution of one sequence of leg transfers).

The *length of step* K during quadruped crawl depends, among others, on the distance of the body's centre of gravity from the ground and on the slope of the terrain.

The allowable range of the step length is determined by a set of inequalities in the following generic form:

$$\begin{aligned} K &\leq f_1 (d, x_c, H, M_s, D, DD, L, a) \\ K &\leq f_2 (d, x_c, H, M_s, D, DD, L, a) \\ K &\leq f_3 (L, d, M_s) \end{aligned} \quad (11)$$

where:

f_1, f_2, f_3 - are linear functions of the mentioned variables [4],

d - is distance between the footsteps on the diagonal of the body (e.g. between footsteps of the left foreleg and right hind leg), d is a constant parameter for a given gait and a given animal,

x_c - coordinate of the body's centre of gravity,

H - vertical distance of the centre of

gravity from the ground (height of the machine),

D - maximum possible forward/backward shift of the leg,

DD - minimum possible forward/backward shift of the body,

L - length of the machine's body,

a - slope of the terrain.

The step length for all of the legs is equal. If we assume that at an instant $t_0 = 0$ the leg touched the ground (it was at its maximum forward shift position - Fig. 8), then up to an constant t ($t \leq r_0 T$) the body moves in relation to this leg by a distance:

- for any of the forelegs

$$d_f = d/2 + x_c' + (2K + 4M_s)/3 + vt \quad (12)$$

- for any of the hind legs

$$d_h = L - d/2 + x_c' + (K + 2M_s)/3 + vt \quad (13)$$

These distances are measured along the direction of motion. If at an instant $t_0 = 0$ the leg was lifted of the ground (it had been at its maximum backward shift position - Fig. 8), then up to an instant t ($t = (1 - r_0)T$) the leg-end moves in relation to the body by a distance S:

- for any of the forelegs

$$S = d/2 + x_c' - (K + 2M_s)/3 + (K + 2M_s)t/(T - r_0 T) \quad (14)$$

- for any of the hind legs

$$S = L - d/2 + x_c' - (K + 2M_s)/3 + (K + 2M_s)t/(T - r_0 T) \quad (15)$$

At a time $t = (1 - r_0)T$ the leg that has been

shifted (has been above the ground), should touch the ground. At this moment the leg as at its maximum forward shift - Fig.8. The relationships (4), (9 -15) are utilised in creation of the description of the machine's gait.

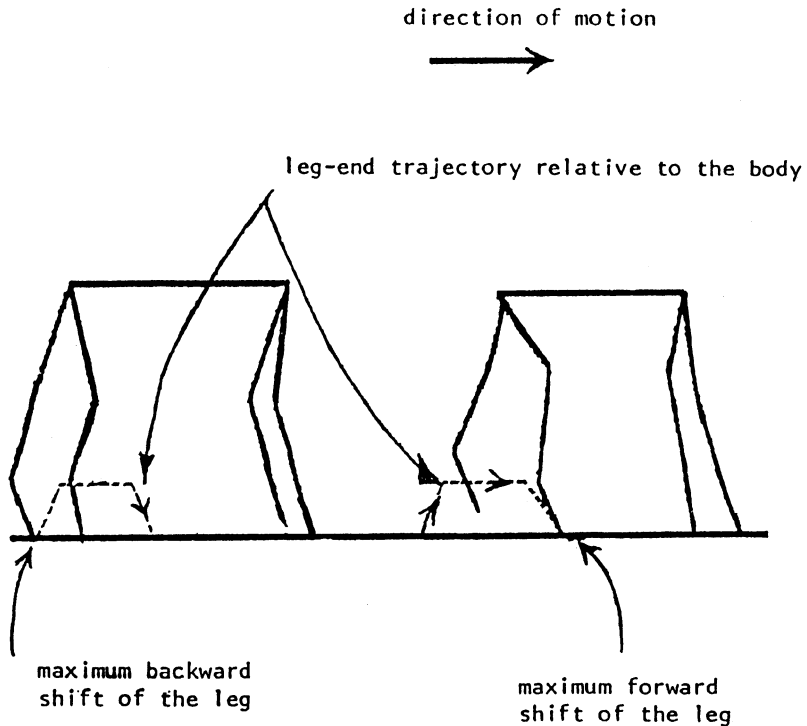


Fig.8. Extremal positions of machine's legs.

4. Results of Investigations

The correctness of the relationships describing the gait model (quadruped crawl) was tested using computer simulation methods. In a similar way the method of modelling the

free gait was tested. We have ascertained that the possible motion-tree search is effective with respect to the speed of data processing.

No flaws have been found in the method (for all of the tested situations the method satisfactory solutions were found). The simulation programs were coded in FORTRAN and executed on an IBM 360/370 compatible computer. The results of simulation trials can be found in [4].

The program modelling the gait of the machine was coded in PASCAL and executed on an IBM/PC microcomputer.

5. Conclusions

The proposed method of gait description and its parameter determination can be used in the case of any statically stable motion. In the currently realised investigation of a walking machine, the utilisation of the results is planned [4]. The method of modelling gait described yields a relationship between a machine's geometric parameters (body length, machine height, the range of motion of each leg) and the gait parameters.

These relationships are instrumental in the design of a machine realising an assumed type of motion or when the types of motion executed by a given machine have to be determined.

The gait description presented in this paper is easier to represent in the structure of a real control system. The computer simulation and last experiments with real machine proves correct the assumed procedure.

The research was supported by the Polish Academy of Sciences

- pr No.7.1.

References

1. Kumar V., Waldron K.,J.,: Gait analysis of walking machines for omnidirectional locomotion on uneven terrain. From preprint submitted for publication, 1988
2. Morecki A., Jaworek K., Pogorzelski W., Zielinska T., Fraczek J., Malczyk G.: Robotics system - elephant trunk type elastic manipulator combined with a quadruped walking machine. Second International Conference on Robotics and Factories of the Future. San Diego 1987
3. Raibert M.H.: Legged robots. Communication of the ACM. 29(6), 499-514 (1986)
4. Zielinska T.: Modelling of the quadruped walking machine gait (in Polish). PhD Dissertation (advisor Morecki A.). Warsaw 1986
5. Morecki A., Busko Z., Jaworek K., Pogorzelski W., Zielinska T., Fraczek J., Malczyk G.: Robotics system: elastic manipulator combined with a quadruped walking machine. Seventh CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators. Udine 1988

Biped Locomotion by FNS: Control Issues and an ANN Implementation

Gideon F. Inbar

Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel.

Abstract. Muscles paralyzed as a result of spinal injury, may be activated to contract with the technique of Functional Neuromuscular Stimulation (FNS). The translation of such externally induced contraction into functional movements of the lower limbs of paralyzed individuals may be considerably improved by the incorporation of a controller into the feedback loop closed around the leg joints. Human locomotion is a complex movement that even with a simplified model exhibits many degrees of freedom, coupling between the joints, nonlinear characteristics in single joint gains and overall system dynamics etc. In the present work a simplified five segment inverted pendulum model is used to demonstrate a joint decoupling scheme that could allow each joint to be controlled independently, using a model reference adaptive controller (MRAC). Linearization around vertical standing position is used for simplification despite the large joints angular displacement during normal locomotion.

MRAC implementation in real time, especially for tracking pre-determined input signals, can be simplified using an inverse dynamics adaptation. An artificial neural network (ANN) implementation of such a scheme, which is based on a perceptron like learning scheme, is outlined. The ANN controller's ability to track the parameters of a highly nonlinear system is demonstrated. Thus the ANN controller can overcome the nonlinearities involved in both the neuromuscular mechanisms and those generated by the mechanical coupling.

It is shown that the network in the adaptation mode can generalize for various inputs, converge from zero initial conditions in less than twenty iterations and maintain performance despite the loss of more than 50% of its synaptic connections.

1. Introduction

Functional neuromuscular stimulation (FNS) is the application of a controlled electrical stimulus to the intact peripheral or intramuscular nerve in an attempt to replace upper motor neuron control which may be lost due to cerebral stroke, brain injury, tumor or traumatic spinal lesion.

Over the past twenty years several FNS orthoses have been developed for the improvement of hand function and gait. Most are open loop while the present

work deals with a closed loop system. Recently various FNS systems have been developed to restrengthen the paralyzed muscles of the lower legs and to enable certain paraplegics to stand and execute a simple gait pattern [1,2,3]. After all these years and great effort in many laboratories and countries the solution to human gait control through FNS is only a distant reality. What are the problems posed by the human body and by the task that we attempt to perform?

There are two different classes of problems: a. Those posed by the task, i.e. to generate a locomotion pattern in the desired progression direction and to simultaneously maintain body stability in space. b. Problems posed by the anatomy and physiology of the body. The latter consist of nonlinearities in the individual muscle gain characteristics, nonstationarities in static and dynamic characteristics in the neuromuscular systems, anatomical and physiological constraints that must be observed – such as the range of motion or torque capabilities, time constants etc. – mechanical coupling of joints through muscles that work across more than one joint, heavy noise in the frequency band of operation due to spasticity, etc.

To overcome the problems posed by the human body, specified here as class (b), we have proposed to use adaptive controllers [4,5], and some modified controller to cope with the system noise [6,7]. In the above references there is a complete outline of the system used, the stimulation parameters, characterization of the static and dynamic properties of the neuromuscular systems used etc. The individual joint controller synthesis is therefore not discussed in the present paper.

In the present paper the more general problem of global controller design is addressed. A brief discussion of biped locomotion model is outlined, mainly in order to emphasize the complexity of the problem at hand.

The first stage of controller synthesis consists of obtaining nominal programmed control inputs from the inverse model of the biped system. The inverse model was given as reference angular trajectories – nominal trajectories – that were measured from healthy slowly walking individuals using an electro-goniometer system [8]. This stage amounted essentially to synthesizing open loop control inputs utilizing *complete system dynamics*, but assuming no perturbations to the system.

The second stage in the design process consists of developing a controller capable of tracking the nominal trajectories. A brief review is given of the design of the state feedback controller for the single support phase that ensures a globally decoupled closed loop system with eigenvalues at desired locations. A somewhat similar approach is used for the double support phase. This and the problems associated with transitions between control modes are not discussed.

A simpler control scheme, more compatible with normal CNS (central nervous system) control of locomotion, had been suggested by us a long time ago [9, 10]. This is basically an inverse dynamic control, where the feedback signals are used to update the inverse model of the plant, i.e. of the legs. Inverse dynamic control through artificial neural network implementation is outlined. The network architecture is given for the special thermometer representation of analog signals [11]. The network performance for nonlinear system dynamics and its ability to generalize for various input signals is demonstrated. Finally the network robustness is studied with up to 70% synaptic failure.

2. The Biped Model and State Feedback Controller for a Single Leg Support Phase

Why use a biped model? Two reasons justify the effort needed to develop such a model.

1. To investigate problems of postural stability and periodic motion of the human body under external control. 2. To use inverse dynamic modeling to generate the required control signals based on the given joint trajectories in slow normal human walking. The concept of inverse dynamic modeling for adaptive neuromuscular control was introduced by us long ago [9] and its application using normal gait [8] for the control of the knee joint in paraplegics was reported recently [6].

Models of human locomotion have been developed before [12]. Such models describe the agonist/antagonist muscle pairs as torque generators acting around a single degree of freedom (DOF), hinge type joints. The human walking gait may involve 20 DOF, a number which is very difficult to handle even with the presently available software tools. We therefore use a simplified compound inverted pendulum of 5 DOF (two lower legs, two upper legs and torso), and movement constrained to the sagittal plane. Biped locomotion is complicated by movement of the structural support point, i.e. the periodic exchange of support from one leg to the other. We describe here only one single support cycle, in order to be brief, and ignore the switching of phases and the double leg support phase.

2.1 The equations of motion for the single leg support phases

The single leg support phases are those sections of the locomotion cycle in which the biped contacts the ground at one point only. Equations were derived using Lagrange's method for an assemblage of interconnected linkages. The positions of the centers of mass of each segment were described with respect to the origin of the coordinate system.

Lagrange's formula was written as

$$M_{q_r} = \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_r} \right) - \frac{\partial T}{\partial q_r} + \frac{\partial V}{\partial q_r} \quad (1)$$

where

- M_{q_r} are the generalized forces
- q_r are the generalized coordinates
- T is the kinetic energy of the system
- V is the potential energy of the system.

Due to the complexity of the system (*5 generalized coordinates*) the equations of motion were derived using the symbolic programming language "REDUCE" [13]. The five D.O.F. model gave five non-linear equations as functions of the angles ($\theta_i, i = 1, 2, \dots, 5$) segment masses, inertia and length parameters as depicted in Fig. 1. The final stage in obtaining the equations of motion involved expressing the generalized moments in terms of the virtual work performed by the non-conservative actuator torques. This then gave the explicit coupling between the D.O.F. expressed as the moment sum at the joints.

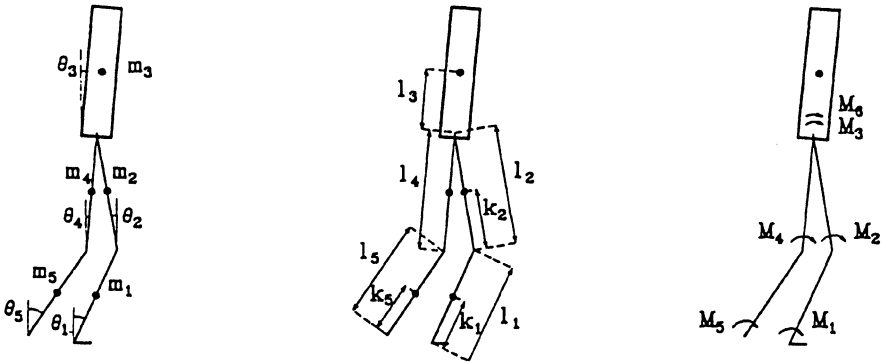
For ease of analysis only linearized equations were used. The non-linear equations were thus linearized with respect to the variables $\theta_i, i = 1, 2, \dots, 5$, and then transformed into the regular state-space representation below.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx}\end{aligned}\quad (2)$$

where

$$\mathbf{x} = \text{state vector} = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, \theta_3, \dot{\theta}_3, \theta_4, \dot{\theta}_4, \theta_5, \dot{\theta}_5]^T$$

and linearization was performed around the state,



$m_1 = 4.55\text{kg}$	$I_1 = 0.105\text{kgm}^2$	$l_1 = 0.502\text{m}$	$k_1 = 0.267\text{m}$
$m_2 = 7.63\text{kg}$	$I_2 = 0.089\text{kgm}^2$	$l_2 = 0.431\text{m}$	$k_2 = 0.247\text{m}$
$m_3 = 49.0\text{kg}$	$I_3 = 2.35\text{kgm}^2$	$l_3 = 0.28\text{m}$	
$m_4 = 7.63\text{kg}$	$I_4 = 0.089\text{kgm}^2$	$l_4 = 0.431\text{m}$	$k_4 = 0.184\text{m}$
$m_5 = 4.55\text{kg}$	$I_5 = 0.105\text{kgm}^2$	$l_5 = 0.502\text{m}$	$k_5 = 0.235\text{m}$

Fig. 1: The five segment biped model.

$$\mathbf{x} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T \quad (3)$$

which, since all angles are referenced to the vertical, represents the biped in an upright posture.

The dynamics matrix **A**, the input matrix **B** and the output matrix **C** are shown in Fig. 2. (Immediately apparent is the fact that the system, in the stance phase, has 6 inputs and 5 outputs).

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 394 & 0 & -383 & 0 & 9 & 0 & 4 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -484 & 0 & 533 & 0 & -40 & 0 & -15 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 24 & 0 & -83 & 0 & 49 & 0 & 10 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -40 & 0 & 127 & 0 & -42 & 0 & -70 & 0 & 23 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 22 & 0 & -56 & 0 & 16 & 0 & 76 & 0 & -56 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Fig. 2: The **A**, **B** and **C** matrices for the linearized equations of motion of the biped in the single leg support phase.

Furthermore, for each of the two single leg support phases, either column 1 or 5 will contain all zeros since only one of the ankle actuator torques may be active at any one time. The stability of the single leg support phases is seen best from a plot of the open-loop pole positions (eigenvalues of the matrix **A**) in the complex s-plane (Fig. 3). These poles are distributed symmetrically about the origin on the real and imaginary axes and show that the biped is highly unstable.

The poles λ_1 and λ_2 represent the entire structure balancing on the lower leg of the support foot. λ_3 and λ_4 may be attributed to that section of the model balancing on the support knee and the outermost poles on the real axis, λ_5 and λ_6 , represent the torso balancing on the entire supporting leg. All these poles are characteristic of a series of inverted pendulums, each balancing on the one below it. The two pairs of complex poles, λ_7, λ_8 and λ_9, λ_{10} are characteristic of regular pendulums i.e. displaying pure oscillatory response, and represent,

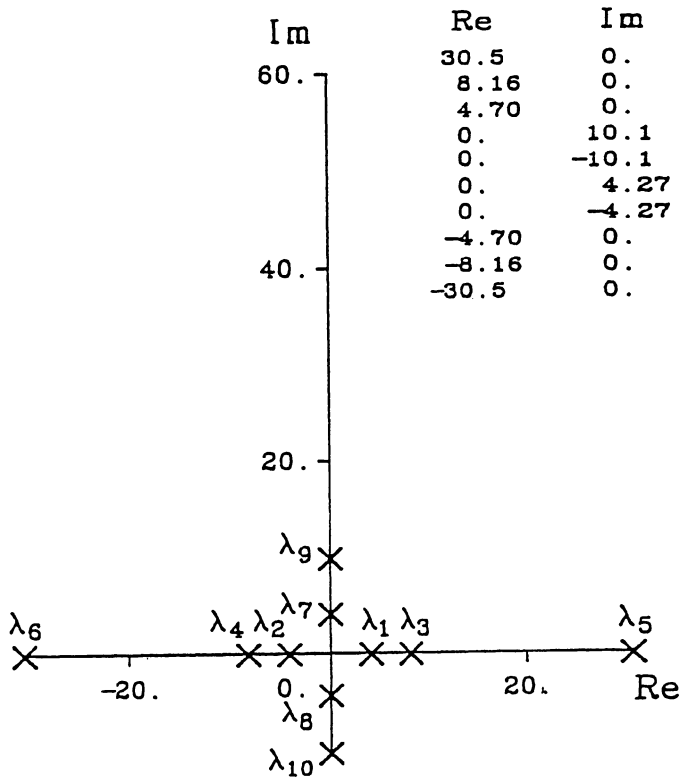


Fig. 3: The open-loop pole positions for the biped single leg support phase.

respectively, the thigh and shank of the swing leg. The open-loop system dynamics are quite obviously unstable and require the addition of feedback for stability. An additional feature of the open-loop system is the coupling between each D.O.F.

2.2 Theoretical considerations in designing the closed-loop system

In linear multivariable (multi-input multi-output) systems, a change in any input will usually result in changes in all output variables. It is essential, for the purposes of simulation and control; to obtain a system in which such interaction between controls does not occur. The design objective of non-interacting (decoupled) systems is therefore to obtain a system in which each input affects only one output. The primary advantage of such a design is that once non-interaction is achieved, the system is reduced to a number of single-input single-output (SISO) channels, or sub-systems, to which well-established controller design techniques

may be applied. It may be seen therefore that two design problems need to be solved. The first is concerned with decoupling the high (10^h) order bipped model into a number of lower order independent sub-systems. The second problem requires designing each independent sub-system such that certain performance criteria are satisfied. The design procedure adopted [14, 15] enabled both these stages to be combined into a single procedure.

The design setting is linear state-space with the mathematics consisting chiefly of matrix algebra. The state is a vector of size 10×1 and the dynamics matrix, a square matrix of dimension 10. The design objective is to obtain a state feedback matrix, $\mathbf{F}$, of dimension 5×10 , such that complete decoupling and pole-placement in each of the independent sub-systems may be achieved. The symbolic programming language "REDUCE" was employed to perform all the symbolic matrix calculations.

The feedback scheme showing the matrices to be designed, $\mathbf{F}$ and $\mathbf{G}$, is shown in Fig. 4. It can be shown that there exist a pair $[\mathbf{F}, \mathbf{G}]$ which will decouple the system. The units of the gains in the feedback matrices are Nm/rad . In Fig. 5 is shown the closed-loop transfer function matrix $\mathbf{R}(s)$ calculated from,

$$\mathbf{R}(s) = \mathbf{C}[s\mathbf{I} - \mathbf{A} - \mathbf{BF}]^{-1}\mathbf{BG} \quad (4)$$

where

$\mathbf{I}$ is a 10×10 identity matrix

Thus we see that a one-to-one correspondence between the input and the output variables has been achieved since $\mathbf{R}(s)$ is diagonal with the off-diagonal elements equal to zero.

The elements on the diagonal of $\mathbf{R}(s)$ represent the transfer functions of each joint subsystem and, since all off-diagonal elements are zero, show that complete decoupling between the subsystems has been achieved. The m_{1i} and m_{oi} ($i=1, \dots, 5$) terms in each individual transfer function, and the individual

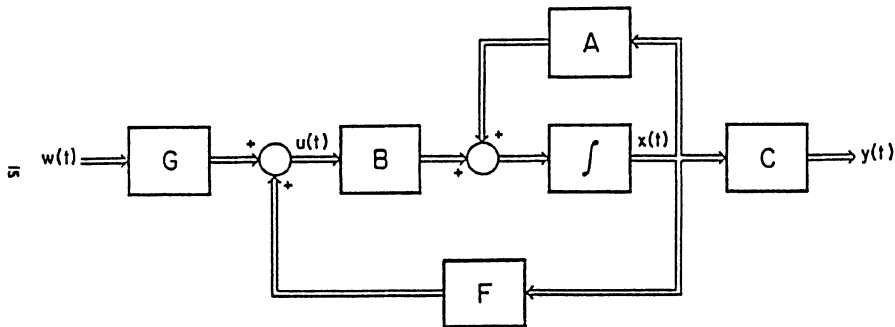


Fig. 4: The state feedback decoupling scheme.

gain γ_{oi} may be chosen to locate the closed-loop poles of each joint subsystem and so meet any desired performance criteria such as critical damping or fast rise time. However freedom in choosing the closed-loop pole positions and gain terms is limited by their direct influence on the magnitude of the feedback gains (elements in the F matrix) and so must be done with the capabilities of each joint actuator in mind.

$$R(s) = \begin{bmatrix} \frac{\gamma_{01}}{s^2 - m_{11}s - m_{01}} & 0 & 0 & 0 & 0 \\ 0 & \frac{\gamma_{02}}{s^2 - m_{12}s - m_{02}} & 0 & 0 & 0 \\ 0 & 0 & \frac{\gamma_{03}}{s^2 - m_{13}s - m_{03}} & 0 & 0 \\ 0 & 0 & 0 & \frac{\gamma_{04}}{s^2 - m_{14}s - m_{04}} & 0 \\ 0 & 0 & 0 & 0 & \frac{\gamma_{05}}{s^2 - m_{15}s - m_{05}} \end{bmatrix}$$

Fig. 5: The closed-loop transfer function matrix ($R(s)$).

2.3 Postural stability and the initiation of gait

The design procedure described in the preceding sections was implemented to provide both complete decoupling between each of the biped's D.O.F. and arbitrary closed-loop pole-placement. In the following, a brief description of the selected closed-loop dynamics is given.

It was decided to maintain the poles representing the swinging leg's dynamics on the imaginary ($j\omega$) axis but to relocate them slightly so as to match more closely the stride length, l_g , and velocity v_g , of a slow periodic gait. The calculated positions were λ_7, λ_8 (thigh dynamics) $= \pm j6$ and λ_9, λ_{10} (shank dynamics) $= \pm j10$. Since the objective was to simulate paraplegic walking, it was decided to shift the poles representing the torso dynamics to the far left hand side of the s -plane in order to rigidly stabilize the upper body in an upright position. This situation resembles closely the state of the upper body in F.N.S. assisted standing of paraplegics. The chosen positions were $\lambda_5, \lambda_6 = -10$ which ensured that $\theta_3 \approx 0$. An F.N.S. assisted, standing paraplegic may introduce disturbances into his dynamic stability by, for example, moving the positions of his arms on supporting parallel bars. It is possible to model these instabilities by inputting random noise sequences to the torso sub-system.

The supporting leg sub-system dynamics were chosen so as to provide optimum damping and reasonably fast correction to angular deviations. To this end, the

knee sub-system poles λ_3, λ_4 were placed at $-3 \pm j3$ while those of the ankle λ_1, λ_2 were placed at -10 and 0 in order to ensure zero steady-state error for inputs up to second-order (ramp). It should be emphasized that because the symbolic form of the system design equations for the matrices F and G was retained, any change in desired closed-loop system dynamics may be very quickly implemented by solving the appropriate quadratic equations in $\mathbf{R}(s)$ (Fig. 5).

Another possible approach to locomotion design involves maintaining the support leg sub-system unstable. This then makes the entire single leg support phase unstable thus ensuring that the biped will fall forward if given the correct initial conditions. By controlling the swinging leg's trajectory, the biped's fall may be "arrested" allowing the double support phase controllers to take over dynamic control of the biped.

Once the system is decoupled, the individual joint control system may be fed with the right vector of signals using the inverse dynamics approach, in which slow gait angular trajectories are fed to the joint plant inverse dynamics. Plant inverse dynamics are then adapted, with those of the controller, based on joint feedback signals. An ANN implementation is described next.

3. Control by Inverse Dynamics Adaptation and an Artificial Neural Network Implementation

Control through inverse dynamics is a rather simple concept [9]. Given that the requested system response is given by a reference model, $T(S)$, when the CNS supply to it an input signal, $U(S)$, find the signal to be supplied to the system, (neuromuscular dynamics), $G(S)$, so as to have the same response $L(S)$

$$L(S) = T(S) \cdot U(S)$$

The system response to a similar input is

$$L'(S) = G(S) \cdot U(S) \neq L(S)$$

A signal conditioner is needed in order to generate a new signal $U'(S)$ such that

$$U'(S) = \frac{T(S)}{G(S)} \cdot U(S)$$

It is clear then that if $U'(S)$ is applied to $G(S)$ its output will be the desired one

$$U'(S) \cdot G(S) = U(S) \cdot \frac{T(S)}{G(S)} \cdot G(S) = U(S) \cdot T(S) = L(S)$$

It is naturally necessary that both $T(S)$ and $G(S)$ be well known and that $T(S)/G(S)$ define a realizable system. It is also obvious that for the case of tracking $T(S) = 1$ and the signal conditioner is the "inverse dynamics" of the

plant. Adaptation by inverse dynamics and its implementation is discussed for example in Widrow & Stearns [20].

We will not elaborate this point any further except stressing the ample evidence that exists today, and that existed long ago [10] that the CNS has the feedback information necessary in order to determine the plant dynamics.

It was previously demonstrated that the plant at hand is of a high order, non-linear and with changing dynamics. Next we examine whether an ANN can identify such a system's inverse dynamics in order to generate the needed control signal.

3.1 ANN architecture for inverse dynamic adaptation

The artificial neural network (ANN) that is used for the inverse dynamic adaptation is not detailed here [11 and 17] except to mention that it is based on a mono-layered network of binary neurons all being fed by the same inputs. The general block diagram of an inverse dynamic controller is shown in Fig. 6 where the adaptive inverse model, both in the identification loop and in the feedforward pathway is performed by the ANN, as shown in Fig. 7. The learning algorithm minimizes explicitly the errors in the output of the NN – defined as the difference between the discretized training signal vector, X_{tr} , and the output of the NN vector driven by the discretized output of the plant. More details of the adaptation learning mode is shown in Fig. 7(a). The NN consists of N binary neurons, N being the number of quantization levels chosen to represent the analog value of the signals in the system.

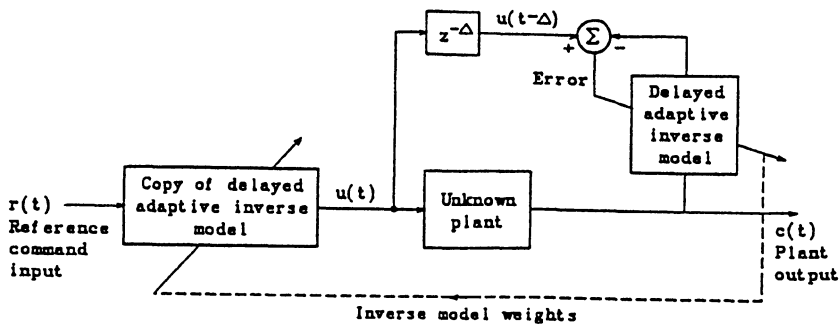


Fig. 6: A block diagram of the inverse model adaptation scheme.

Using vector notation the input-output of these neurons can be written as

$$U_i(n) = W_i \cdot I(n) - T_i \quad ; \quad X(n) = g \{ U(n) \} \quad (5)$$

where W_i is a row vector whose j -th component W_{ij} corresponds to the strength of synaptic connection between the j -th entry of the binary input state vector at time n , $I(n)$, to the i -th neuron, T_i is the threshold of the i -th neuron, X is the output state vector, U is the post-synaptic potential vector and $g(\cdot)$ is applied on an element by element basis.

$I(n)$ consists of a set of delayed inputs, $Y(n) \cdots$, that evolve from the unknown plant driven by the training signal and the delayed outputs, $X(n-1) \cdots$, of the NN itself, as seen in Fig. 7(a) — $Y(n)$ here is $C(t)$ of Fig. 6. The feedback and delays are essential properties of this scheme that enable it to model the unknown discrete dynamic systems. The number of delays should estimate the order of the plant, and this is the only information about the plant needed to be known a priori — or it can be estimated successively by starting with a larger number of delays.

The learning algorithm: The problem is defined as finding the synaptic weights, W_{ij} , and thresholds, T_i , that realize the mapping of the input training vector $I_{tr}(n)$ to the NN output vector $X_{tr}(n)$, $n = 1, \dots, L$. One of the ways to solve the problem is by the iterative learning algorithm based on the perceptron learning scheme [17].

1. **Initialization:** Choose arbitrary $W_{ij}(0)$, $T_i(0)$.

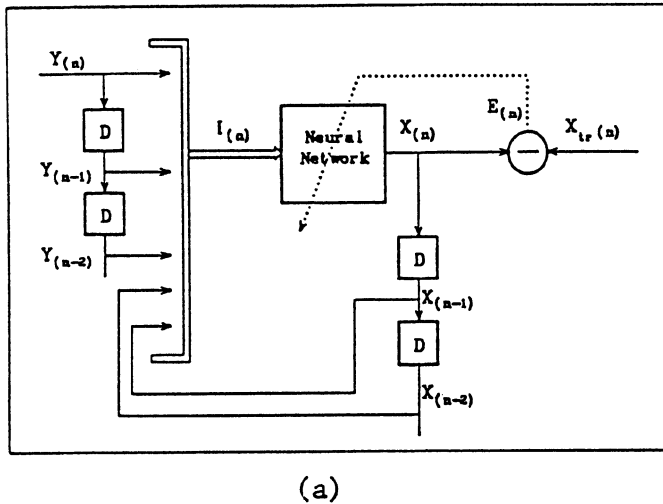
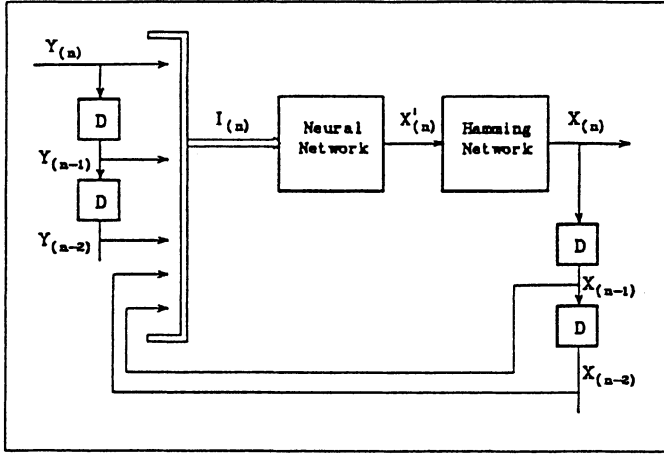


Fig. 7(a): The ANN architecture in adaptation mode.



(b)

Fig. 7(b): The ANN architecture in control mode.

2. *The k -th iteration:* Measure for each $I_{tr}(n)$, $n = 1, \dots, L$ the output $X_i(n)$, $i = 1, \dots, M$

$$X_i(n) = g \{ W_i(k) \cdot I_{tr}(n) - T_i \} \quad (6)$$

Define the error

$$E_i(n) = X_{tri}(n) - X_i(n) ; \quad i = 1, \dots, N \quad (7)$$

Update the weights and thresholds

$$\Delta W_i = \eta \cdot E_i \cdot I_{tr}(n) \quad (8)$$

$$\Delta T_i = -\eta E_i \quad (9)$$

η is the step size of the algorithm. Note that the parameters W_{ij} and T_i are being updated L times during each iteration, where L is the number of transitions in the training set. It can be shown that this algorithm converges if a solution exists [17] or when it does not exist a gradient descent search is performed to minimize a certain cost function [18].

Signal representation: A thermometer presentation of the signals is used in the present work. Signal representation in an ANN is an important issue. The thermometer representation, or threshold decomposition representation [21], was selected due to its distributive characteristics, i.e.

the number of bits equals the number of quantization levels (analog levels). It has the additional important property that each bit in this presentation has the same weight – not like the usual binary or “mother cell” representations. This allows for a simple self-correction network to be incorporated in the control mode, as shown in Fig. 7(b).

3.2 The control mode with ANN

In the control mode the ANN act as an inverse of the plant. The learning of the plant by the ANN is carried out with a training signal and it is therefore imperative that the network is able to generalize, or learn to respond correctly, to input sequences it has not especially been trained for. Because of the particular signal representation used it is possible to improve the networks generalization ability by introducing a pre-programmed error-correcting network. Such a network can be a content addressable memory (CAM) of the Hopfield type [15], where the correct thermometer representation binary vectors are the stable states. Instead a feedforward layered NN programmed to act as a Hamming net [15] was selected. Each time the NN output is not one of the thermometer binary vector representations, the Hamming network will produce a valid thermometer representation vector that is the closest to the output state vector in the Hamming distance sense. The system in a control mode is shown in Fig. 7(b).

Simulation results: In a series of computer simulations the learning and generalization abilities of the NN model identifying the inverse dynamics of linear and nonlinear plants were examined. A nonlinear plant dynamics described by

$$y(n) = [0.7y(n-1) - 0.75y(n-2)]^3 - x^3(n) \quad (10)$$

was used to show the performance of the system. N , the number of quantization levels and number of neurons, was selected to be 31. The length of the training signal, a white noise evenly distributed random noise, was 200. The improvement in tracking the random training signal with an increased number of learning iterations is shown in Fig. 8. In (a) the training sequence is shown in a bold line and the ANN output before training is shown in the dotted line. In (b) the dotted output is achieved after 20 training iterations. The generalization power of the network is shown in (c) where the network is tracking a new random sequence. The ability of the network to track a periodic signal is shown in (d). The network was checked for robustness by randomly eliminating synapses (letting a certain percentage of W 's go to zero). With 30% synaptic loss the network performed well (the generalization errors doubled), however it almost regained its initial performance after retraining.

The complexity of the plant should be kept in mind when judging the results. The ANN architecture proposed here was shown to have two important properties, i.e. ability to generalize and robustiveness. This architecture can be easily implemented in a special purpose hardware in order to function in real time. The major

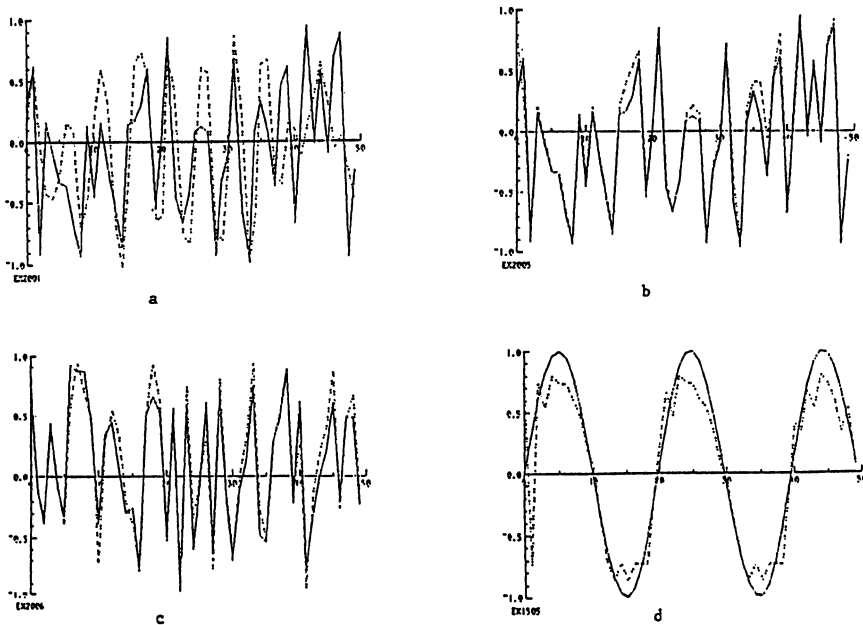


Fig. 8: Performance of an ANN with 31 neurons, two delays on the external input and one on the state vector. Length of the training set is 200 (see text). (a) Before training. (b) After 20 iterations. (c) Generalization with a random sequence. (d) Tracking of periodic signals.

draw back of this architecture is the separation between the learning and identification phase and the control phase. This separation makes also for the inferior tracking ability. What is needed is a scheme that will combine the two phases so that the learning will be carried out simultaneously with control. In such a scheme the ANN synaptic weights will be continuously updated to fit the load and input signal. This will improve tracking accuracy, especially in the face of changing plant parameters. Such a scheme has recently been implemented [21].

4. Conclusions

Walking with an FNS orthotic system was shown here to pose some major control problems, i.e. complexity in dimensionality and strong interaction between joints, nonlinearities and nonstationarities in system parameters. The decoupling scheme proposed here works well in theory, however in reality a complete and continuous decoupling is hard to achieve due to the nonstationarities that the system exhibits. The ANN adaptive scheme presented here can cope with the nonlinearities of the system and long term nonstationarities, and thus provides a solution to some of the inherent difficulties in controlling locomotion by FNS.

Acknowledgement. This work was supported by The Foundation for Research in Electronics, Computers and Communication, Administered by The Israel Academy of Science and Humanities.

References

1. Kralj, A., Bajd, T., Turk, R., Krajnik, J. and Benko, H.: "Gait Restoration in Paraplegic Patients: A Feasibility Demonstration using Multichannel Surface Electrode F.E.S.", *Journal of Rehabilitation Research and Development*, Vol. 20, No. 1, pp. 3-20, Veterans Administration, 1983.
2. Braun, Z., Mizrahi, J., Najenson, T. and Graupe, D.: "Activation of Paraplegic Patients by Functional Electrical Stimulation: Training and Biomechanical Evaluation", *Scand. J. Rehab. Med. Suppl.*, Vol. 12, pp. 93-101, 1985.
3. Chizeck, H.J., Lalonde, R., Chang, C.W., Rosenthal, J.A. and Marsolais, E.B.: "Performance of a Closed-Loop Controller for Electrically Stimulated Standing in Paralyzed Patients", *Proc. 8th Ann. RESNA Conf.*, Memphis, Tennessee, 1985.
4. Goodwin, G.C. and Sin, K.W.: *Adaptive Filtering Prediction and Control*, Prentice-Hall Inc., 1984.
5. Allin, J. and Inbar, G.F.: "FNS Control Schemes for the Upper Limb", *IEEE Trans. Biomed. Eng.*, Vol. BME-33, No. 9, pp. 809-817, September 1986.
6. Hatwell, M.S., Oderkerk, B.J. and Inbar, G.F.: "A Model Reference Adaptive Controller to Control the Knee Joint of Paraplegics", *ICCON*, April, Jerusalem, 1989.
7. Hatwell, M.S., Oderkerk, B.J. and Inbar, G.F.: "The Development of a Model Reference Adaptive Controller to Control the Knee Joint of Paraplegics", *IEEE Trans. on AC*, Vol. 36, No. 6, pp. 683-691, June 1991.
8. Oderkerk, B.J. and Inbar, G.F.: "Walking Cycle Recording and Analysis for FNS Assisted Paraplegic Walking", *MBEC*, Vol. 29, No. 1, pp. 79-83, 1991.
9. Inbar, G.F. and Yafe, A.: "Parameter and Signal Adaptation in the Stretch Reflex Loop", *Progress in Brain Res.*, Vol. 44, pp. 317-337, 1976.
10. Inbar, G.F.: "Muscle Spindles in Muscle Control: Analysis of Adaptive System Model", *Kybernetik*, Vol. 11, pp. 130-141, 1972.
11. Levin, E., Gewirtzman, R. and Inbar, G.F.: "Neural Network Architecture for Adaptive System Modeling and Control", *Neural Networks*, Vol. 4, pp. 185-191, 1991.
12. Hemami, H. and Farnsworth, R.L.: "Postural and Gait Stability of a Planar Five Link Biped by Simulation", *IEEE Trans. Auto. Cont.*, pp. 452-458, June 1977.
13. Hearn, A.C.: "REDUCE - User's Manual Version 3.2", *Rand Publication No. CP78*, The Rand Co., Santa Monica, Calif., April 1985.
14. Wolovich, W.A.: "Linear Multivariable Systems", *Appl. Mathematical Sciences*, Vol. 11, Springer Verlag 1974.
15. Sinha, P.K.: "Multivariable Control, An Introduction", *Electrical Eng. and Electronics*, Vol. 19, Marcel Dekker Inc., 1984.
16. Minsky, M. and Papert, S.: "Perceptrons", MIT Press, 1969.
17. Levin, E.: "A Recurrent Neural Network: Limitations and Training", accepted for publication in *Neural Networks Journal*, 1988.

18. Hopfield, J.J.: "Human Memory Error Correcting Codes and Spin Glasses", *Proc. Natl., Acad. Sci., U.S.A.* 79, 1982.
19. Lippman, R.P.: "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, April 1987.
20. Widrow B. and Stearns, S.: "Adaptive Signal Processing", Prentice-Hall, 1985.
21. Gewirtzman, R. and Inbar, G.F.: "ANN Adaptive Controller using Threshold Decomposition Architecture". In preparation.

How Fast Can a Legged Robot Run?

Jeff Koechling¹ and Marc H. Raibert²

¹ Sibley School of Mechanical and Aerospace Engineering
Cornell University, Ithaca, NY 14853-7501, USA

² Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract. Several parameters can limit the running speed of a legged system. Among them are the strength, length, and stiffness of the legs, the range of joint motion, and the actuator force-velocity characteristics. We have explored how varying these parameters affects top running speed. We developed a dependency tree that suggests that a robot should have long, strong, stiff legs, and actuators with high peak velocity in order to run fast. We have also proposed three ways to improve the control of body attitude in high speed running: keeping hip motions symmetric, compensating for actuator characteristics, and accelerating the hip joint in anticipation of touchdown. In laboratory experiments a planar two-legged robot has reached a top speed of 5.9 m/s (13 mph).

Keywords. Running, Locomotion, Speed.

1 Introduction

“How fast can it go?” Since time immemorial, people have staged races to compare the speed of people, animals, or vehicles. Speed excites people, often because it is the focus of a competition, and sometimes because of the danger or novelty of going fast. Speed is also an easily understood measure of performance. It summarizes the capabilities of a complex system with a single number that says something about an athlete’s prowess, an animal’s likelihood of survival, or a vehicle’s utility.

Three things limit the speed of a vehicle: the power available to overcome drag, the ability of the structure to withstand loads, and the stability of the motion in the face of disturbances. As the system accelerates it encounters a limit of power, strength, or stability that establishes its maximum speed. If the limitation is power, then at maximum speed the drag force cancels the thrust force, leaving no thrust to accelerate the system. If the limitation is strength, then at maximum speed the loading on some component equals its strength, and any increase in speed would cause it to break. If the limitation

is stability, then at the maximum speed some equilibrating mechanism is at its stability limit, and at any higher speed the system would tumble out of control. Most vehicles are designed so that their speed is limited by power, rather than by structure or stability, since a simple inability to accelerate is preferable to structural failure or loss of control.

How fast a legged system can run depends on its design, and on how it is controlled. The important parts of the design are the legs and the hips. To run fast, the legs should be long, strong, springy, and stiff, and the hips should be able to rotate rapidly and through a large angle. The control system must coordinate the actions of the legs and hips so as to regulate the momentum of the system in the horizontal, vertical, and rotational directions. The principles of symmetry, modeling, and anticipation help the control system to regulate rotation of the body.

Running speed is the product of step length and step frequency. The simple prescription for fast running is to take long steps, and to take them quickly. Step length depends on leg length and hip joint range of motion, while stepping rate depends on leg stiffness, leg strength, and hip rotation rate.

Leg length — Long legs allow long steps, so legs should be long for fast running. The distance that a legged system can move forward while its foot is on the ground is proportional to how long its legs are. Mass, moment of inertia, and strength all depend on leg length, and there is a limit to how big a leg can be and still be strong enough to support its own weight and the inertial forces required to move it.

Hip joint rotation — The distance that a legged system can travel forward during a bounce depends not only on the leg length, but also on the range of motion of the hip joint. The hip joint limits how far the leg can pivot during stance without disturbing the attitude of the body.

Hip rotation rate — Hip rotation rate can directly limit running speed. During stance, a legged system is like a polar manipulator. The foot remains fixed on the ground, and the leg length and hip angle determine the position of the body with respect to the foot. The hip position, plus the rate of extension of the leg and the rate of rotation of the hip joint determine the velocity of the body. The faster the hip joint can rotate, the faster the body can advance during stance.

Leg stiffness — A running system alternately bounces off of the ground and flies through the air. For the system to bounce, the legs must be springy. During each bounce against the ground, ground contact forces reverse the vertical momentum of the system. As the legs compress during stance, they build up force, and the vertical component of that force reverses the vertical momentum of the system. The stiffer the springs are, the faster the forces build up and the more quickly the system bounces.

Leg strength — The total impulse required to reverse the vertical momentum of the body is the integral of the contact force over the duration of

the bounce. The legs must be strong enough to transmit the ground reaction force to the body without breaking or buckling. The shorter the bounce, the larger the contact forces must be. Thus, the stronger the legs are, the faster the system can bounce off of the ground without damaging the legs.

Symmetric leg motions — By moving its two legs symmetrically, a biped minimizes how much its body attitude deviates from the nominal angle. If the hip joints are at the center of gravity of the body, then the only disturbances to the body attitude are caused by hip torques. Equal and opposite motions of the hip joints ensure that the hip torques cancel out, and thus do not disturb the body attitude.

Actuator velocity compensation — During stance, the hip of the stance leg is pushed forward by the body, causing the hip to rotate at a rate proportional to running speed. Velocity dependent torques in the hip joints should be compensated so that the body is not rotated forward with the leg.

Ground speed matching — When the leg touches down, an impulsive contact force brings the unsprung mass of the foot to rest. At high speeds, this impulse is not aligned with the axis of the stance leg, so it tends to rotate the leg. The impulse happens very quickly, faster than the hip joint can respond, so some torque is transmitted to the body, which also begins to rotate. If the control system anticipates the touchdown, and accelerates the hip joint before impact, then the impulse is aligned with the axis of the leg. In this case the hip joint does not transmit any torque to the body.

Taken together, symmetric leg motions, actuator velocity compensation, and ground speed matching substantially reduce body attitude disturbances associated with running fast.

In the following section we review relevant previous studies. Then we develop a dependency tree that expresses the speed of a running system in terms of its physical parameters. Finally, we present laboratory experiments suggested by the analysis.

2 Background

The study of running is interdisciplinary. Some areas of research that provide results helpful in understanding running speed are:

- Studies of running animals
- Creation of artificial legged systems (robots)
- Investigation of the performance of vehicles in general

Biologists have studied innumerable aspects of running animals, including their structure, the motions and forces that occur during running, and the energy consumed for different speeds and gaits. Robots and vehicles that

travel on legs provide a way to study walking and running in simple, easily instrumented systems, without the complexity inherent to biological systems. Research into the performance of boats, aircraft, and land vehicles is relevant to studying running speed, because the task of locomotion and the physical principles of support, balance, and progress are common to all vehicles.

2.1 Biomechanical Research

2.1.1 Scaling

Biologists have proposed various similarity models to describe how the shape, structure, and motions of an animal depend on its size. Similarity models either describe measurements gathered from animals that are similar in arrangement but vary in size, or they describe the way that such measurements should vary in order to keep some quantity the same at all sizes.

Similarity models apply to all mechanisms, not just biological ones. A mechanism design only works over a limited range of sizes. Typically, at a very small size the ratio of viscous forces to inertial forces increases and the resultant damping prevents the mechanism from operating. At a very large size, gravitational forces become dominant and exceed the strength of the materials.

Hill (1950) concluded that speed is independent of size for animals of similar design. An animal makes movements that are proportional to its size, but at a frequency that is inversely proportional to its size. For geometrically similar animals, the differences cancel out, so top speed is the same regardless of size. Hill introduced the idea of physiological time, saying that animals live on a time scale proportional to body size. Thus large animals live longer than small animals, their hearts beat more slowly, and they take more time for each running step. Hill also noted that for large animals, a greater portion of skeletal and muscular strength is required to support the animal's weight than for small animals.

McMahon (1975) compared and discussed three scaling laws: geometric similarity, elastic similarity, and static stress similarity. Geometrically similarity preserves shape across scale, as all linear dimensions change with the same scale factor. Elastic similarity preserves resistance to column buckling. For columns of different size to have the same safety factor against buckling, long columns must be relatively thicker than short columns. The scale factor for the diameter of elastically similar columns is $3/2$ the scale factor for length. Static stress similarity preserves resistance to bending failure in simply supported beams bearing their own weight. In this case, the scale factor for diameter is twice the scale factor for length.

Comparison of Three Similarity Models

	geometric similarity	elastic similarity	static stress similarity
length, ℓ	$\ell \propto W^{1/3}$	$\ell \propto W^{1/4}$	$\ell \propto W^{1/5}$
diameter, d	$d \propto W^{1/3}$	$d \propto W^{3/8}$	$d \propto W^{2/5}$
surface area, S	$S \propto W^{2/3}$	$S \propto W^{5/8}$	$S \propto W^{3/5}$
cross sectional area, A	$s \propto W^{2/3}$	$A \propto W^{3/4}$	$A \propto W^{4/5}$
natural frequency, ω	$\omega \propto W^{-1/3}$	$\omega \propto W^{-1/8}$	$\omega \propto W^0$
speed, V	$V \propto W^0$	$V \propto W^{1/8}$	$V \propto W^{2/5}$

Table 1: Three similarity principles predict different variations in shape and speed as a function of body weight, W . The table is adapted from a paper by McMahon (1985), who cites examples of animal measurements that match the elastic similarity model.

McMahon presented a variety of evidence that the design of animals is in accordance with elastic similarity. In particular, the running speed of animals is proportional to body weight raised to the $1/4$ power, as predicted by elastic similarity. Geometric similarity predicts an exponent of zero, and static stress similarity predicts an exponent of $2/5$. Table 1 includes a few predictions of the three similarity models.

Alexander (1988) found that animals weighing over 20 kg scale according to elastic similarity, while the dimensions of smaller animals obey geometric similarity. Alexander also describes an extension of geometric similarity called dynamic similarity. For geometric similarity, animals of different sizes undergo a uniform scaling of linear dimensions. For dynamic similarity, the scaling of linear dimensions is accompanied by uniform scaling of time and force. Thus, dynamic similarity specifies not only the change in shape of animals at different sizes, but also changes in their motions. Alexander proposes that the scale factor for forces be the cube of the scale factor for length, and that the scale factor for time be the square root of the scale factor for length. These scale factors work for motions characterized by gravitational and inertial forces. However, Alexander points out that if both elastic and gravitational forces are important it is impossible to maintain strict dynamic similarity.

2.1.2 Energetics

The power required for running increases with speed. The environment opposes motion with drag forces, dissipating power equal to the product of the forward speed and the drag force. Drag forces remain constant or increase with running speed. For example, the gravitational drag caused by climbing

a hill is independent of speed, while aerodynamic drag increases as the square of running speed.

Several investigators have measured the oxygen consumption of running animals. Consumed oxygen produces metabolic energy at a rate of $1 \text{ ml O}_2 = 20.1 \text{ J}$. The rate of oxygen consumption indicates the metabolic power produced by the animal, some of which is used to overcome the resistance of the environment, some of which is dissipated in muscle inefficiency, and some of which is used to maintain the animal's basal metabolism.

The rate of metabolic energy consumption increases with running speed. Taylor, Heglund, and Maloiy (1982) report that the energy consumed during running is:

$$\dot{E}_{\text{metab}}/M_b = 10.7M_b^{-0.316}v_g + 6.03M_b^{-0.303} \quad (1)$$

where E_{metab} is the metabolic energy consumed in watts, M_b is the body mass in kg, and v_g is the running speed in m/s. The equation is based on measurements from 60 species of animals, ranging in size from 0.0072 kg pygmy mice to 254 kg zebu cattle. It indicates that energy consumption increases linearly with running speed, and that the rate of increase is smaller for large animals than for small animals.

Dawson and Taylor (1973) studied the energetic cost of locomotion in kangaroos. They found that over a range of speeds from 2 m/s to 6 m/s, hopping frequency and energy consumption remain nearly constant, and the stride length increases in proportion to the speed. This is in contrast to the linear increase in energy consumption with speed indicated by equation 1. Dawson and Taylor attributed the constant consumption of energy with increasing speed to increased storage and recovery of elastic energy, particularly in the kangaroo's large Achilles tendon.

Alexander and Vernon (1975) measured the ground forces exerted by hopping kangaroos using force plates, and combined the measurements with film records of the motion to determine the fluctuations of energy during hopping. They calculated that elastic storage of energy in the kangaroo's Achilles tendon reduced the energetic cost of hopping by 40%. Alexander and Vernon also noted that the kangaroo's tail rotates in the opposite direction from the legs, in a way that reduces the angular motions of the body. Large sheets of elastic tendon along the tail contribute to its oscillation.

McMahon and Greene (1978, 1979) considered not only the compliance of a runner's legs, but the compliance of the ground as well. Their model predicted that top running speed would be slightly greater on a compliant track than on a rigid surface. The model predicted that the fastest running would be on a track four times as stiff as the runner's legs. McMahon and Greene built such a track at Harvard University, and observed the predicted 2% increase in running speed, along with a decrease in injuries. This study points out the importance of the environment in determining the top speed of a running system.

Legged Robots and Vehicles

Machine	Leg Length m	Speed m/s mph	
6 legs			
OSU Hexapod (McGhee)	0.8	0.3	0.7
USSR Hexapod (Gurfinkle)	0.35	0.1	0.2
SSA Hexapod (Sutherland)	1.0	0.14	0.31
ODEX (Odetics)	1.3	0.5	1.1
ASV (Waldron)	1.9	2.2	5.0
4 legs			
PV II (Hirose)	0.87	0.5	1.1
Quadruped (Raibert)	0.66	3.0	6.7
2 legs			
WL10-RD (Kato)	0.96	0.23	0.51
Biper-3 (Miura)	0.20	0.02	0.04
MEG-2 (Funabashi)	0.48	0.5	1.1
Kenkyaku (Furusho)	0.72	0.8	1.8
Planar Biped (Koechling)	0.8	5.9	13.1

Table 2: These are a few of the machines and vehicles that have been used to study the control of legged locomotion. None of the machines was explicitly designed for high speed. All of the leg lengths and speeds are approximate.

Hoyt and Taylor (1985) studied the energetics of horses walking, trotting and galloping. For each gait they found one speed that provided the best energy efficiency, and if allowed to move freely, a horse always chose a speed and gait that corresponded to a local maximum in efficiency.

2.2 Robotic Research

There are very few artificial systems that walk or run. Walking and running require dynamic stability, meaning that the system moves continuously in order to keep the average point of support beneath the center of gravity. Table 2 lists the number of legs, leg length, and speed of several legged robots. Keep in mind that none of these systems was designed for speed. Each was designed to study some aspect of the control of locomotion.

Huang and Waldron (1987) derived the relationship between weight and maximum speed for a hexapod vehicle crawling with a particular gait. By assuming that the distribution of forces among the support legs was a linear function of the forward and lateral position of the legs, and requiring the vehicle to remain in static equilibrium, they determined the proportion of the vehicle's weight on each leg as a function of speed. By limiting the force

on the most heavily loaded leg to the maximum safe load they computed the tradeoff between speed and payload.

2.3 Vehicular Research

Gabrielli and von Kármán (1950) studied the cost of locomotion at different speeds. They gathered data on the gross weight, installed power, and maximum speed of many land vehicles, ships, boats, aircraft and animals. For each vehicle they computed the *specific resistance*, which is the ratio of power to the product of weight and velocity: $\epsilon = P/WV$. This nondimensional quantity is a measure of the energetic cost of locomotion. Gabrielli and von Kármán's data indicate that any particular means of locomotion is only energy efficient over a narrow range of speeds, and that in general small, fast vehicles are less efficient than large, slow vehicles. For example, a merchant ship has a specific resistance of about 0.003 at a speed of 6 m/s, while a jet fighter plane has a specific resistance of 0.3 at a speed of 300 m/s.

The graphs of specific resistance as a function of speed show a limiting line, a minimum specific resistance that increases with speed. This line represents an efficiency limit imposed by aerodynamic or hydrodynamic drag. Single vehicles of sufficient size should have specific resistances below the limit, because specific resistance decreases with vehicle size. Railroad trains achieve a greater energetic efficiency because the aerodynamic drag is smaller for the train than for the individual cars, and because the power is supplied by a few large, efficient engines. Gabrielli and von Kármán point out that the size of the vehicles that can be built is limited by the strength to weight ratio of the available construction materials.

3 The Dependency Tree

One way to impose structure on the relationship between the physical parameters of a mechanism and how fast it can run is to build a dependency tree, as shown in figure 1. The top of the tree is running speed. The branches are formed by expressing running speed as the ratio of step length to step period, and successively refining those quantities to simpler characteristics of the running motion. The leaf nodes are parameters of the links, joints, and actuators. Body attitude is an exception; it is determined by how well the control system corrects disturbances.

The equations in figure 1 embody several assumptions, which are explicitly stated in the paragraphs below. The resulting analysis accurately represents the kinematics of the mechanism, but it uses simplified dynamics, and says nothing about energetics. It leads to tractable expressions for running speed that qualitatively predict how parameter variations affect speed.

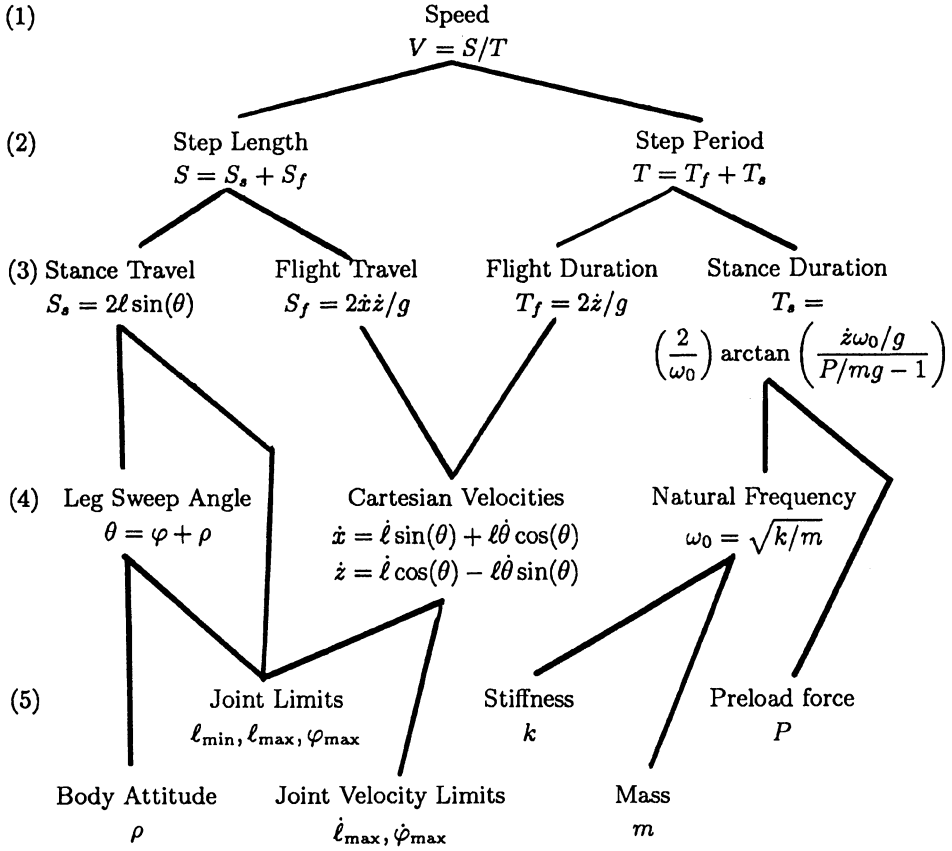


Figure 1: A tree indicating some of the ways the running speed of a legged system depends upon parameters of the mechanism. The top of the tree represents speed, a measure of performance. The intermediate rows represent characteristics of the running motion, and the bottom row represents the parameters of the mechanism. The state variables of the system are the leg length, the rate of leg extension, the leg angle, and the rate of leg rotation. The operating point is described by the state at the moment of liftoff: ℓ , $\dot{\ell}$, θ , and $\dot{\theta}$. The horizontal and vertical components of liftoff velocity are $\dot{x}$ and $\dot{z}$. If the steps are symmetric and the pitch angle of the body is uniformly zero, then the operating point completely characterizes the motion. Steps are symmetric if the horizontal velocity and the leg length are the same at touchdown as at liftoff, and the vertical velocity and the leg angle change signs from touchdown to liftoff. The coordinate system is shown by figure 2.

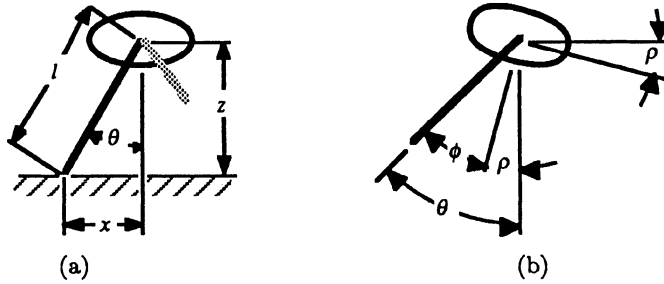


Figure 2: (a) The horizontal and vertical distances from the foot to the hip are determined by the leg length and leg angle: $x = \ell \sin(\theta)$, $z = \ell \cos(\theta)$. During stance the foot is motionless, so the derivatives of the hip coordinates give the horizontal and vertical velocity with respect to the ground: $\dot{x} = \dot{\ell} \sin(\theta) + \ell \dot{\theta} \cos(\theta)$, $\dot{z} = \dot{\ell} \cos(\theta) - \ell \dot{\theta} \sin(\theta)$. (b) The leg angle is the sum of the leg angle with respect to the body, and angle of the body with respect to the ground: $\theta = \phi + \rho$. The three angles, θ , ϕ , and ρ are measured clockwise from the nominal position, in which the leg is vertical and the body horizontal.

3.1 Speed

The first row of the dependency tree in figure 1 is the definition of running speed (V), which is the ratio of the forward progress on each step (S) to the time required to complete that step (T):

$$V = S/T. \quad (2)$$

To increase its speed, a running system must take longer steps, more frequent steps, or both. In bipedal running, stance and flight proceed in strict alternation, and a step consists of exactly one stance and one flight. Figure 3 shows that the distance traveled during one step is the sum of the distance traveled during stance (S_s) and the distance traveled during flight (S_f):

$$S = S_s + S_f. \quad (3)$$

Likewise, the time required for a step is the sum of the stance duration (T_s) and the flight duration (T_f):

$$T = T_s + T_f. \quad (4)$$

These definitions make up the second row of the tree.

In the third row of the tree, the dynamics of the system come into play. In the stance phase, the system resembles both a mass bouncing on a spring and an inverted pendulum pivoting over its fulcrum, as shown in figure 4. The dynamics are much simpler during flight, when the system approximates a rigid ballistic projectile rising and falling under the influence of gravity.

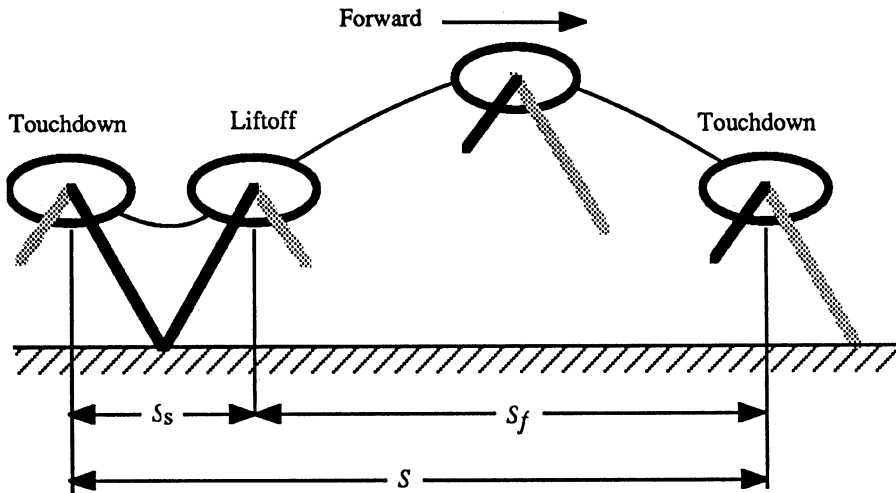


Figure 3: Step length (S) is the sum of the forward progress of the hip during stance (S_s) and the forward progress of the hip during flight (S_f).

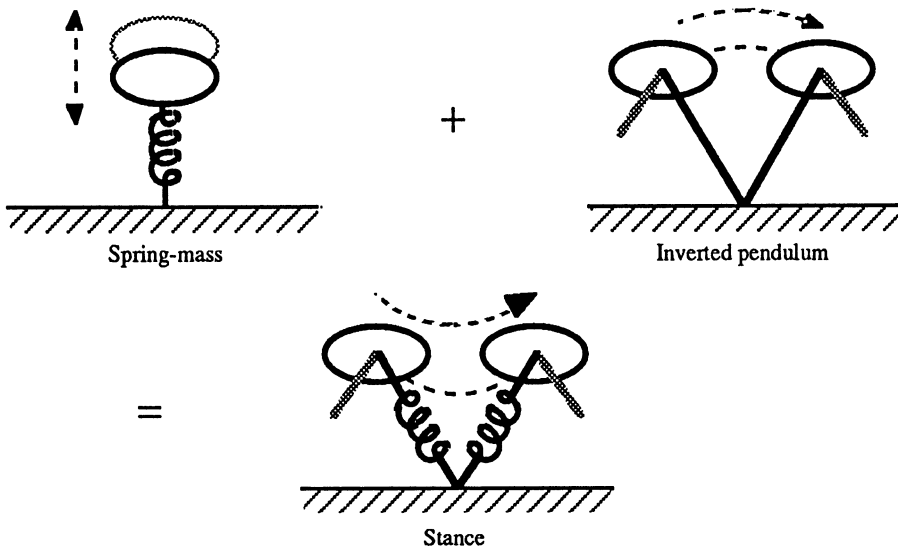


Figure 4: The motion of a running system during stance results from the interaction of two simpler motions. The vertical motion is predominantly the bouncing motion of a spring-mass oscillator. The forward travel results from the tipping motion of an inverted pendulum that moves first toward and then away from the unstable equilibrium point. Forward speed decreases during the first half of stance, because some of the horizontal kinetic energy is temporarily stored in the leg spring. During the second half of stance, the spring releases energy and the system speeds back up.

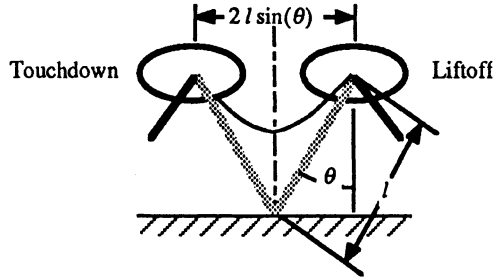


Figure 5: The distance traveled during stance is a function of the leg length and hip angle at the beginning and end of stance. If the motion is symmetric, so that the state at liftoff is a mirror image of the state at touchdown, the distance traveled is $2\ell\sin(\theta)$. The system in the figure is moving from left to right, touching down with the foot in front of the hip and lifting off with the foot behind the hip.

3.2 Stance

The leftmost and rightmost nodes of row three describe the stance duration and the stance travel. The stance travel is the forward progress of the body during stance, and is a function of the leg lengths and leg angles at the beginning and end of the stance phase, as shown in figure 5. For symmetric steps, the leg length and leg rotation rate are the same at touchdown and at liftoff, while the leg extension rate and the leg angle have opposite signs at touchdown and at liftoff. Under the assumption of symmetry, the distance traveled during stance is:

$$S_s = 2\ell\sin(\theta), \quad (5)$$

where θ and ℓ represent the leg angle and leg length at liftoff.

The motion during stance is described by a pair of coupled second-order non-linear differential equations. The stance duration can be computed by integrating these equations forward in time from the moment of touchdown until the moment of liftoff. We know of no closed form expression for the stance duration as a function of the mechanism parameters and the state of the system at touchdown. In order to proceed with the analysis, we pretend that the horizontal and vertical motion of the system are decoupled, and that the stance duration is determined only by the vertical motion.

For the simple case of a mass bouncing vertically on a linear spring, McMahon (1986) showed that the time required to rebound from the ground depends on a parameter he called the Groucho number. Modifying McMahon's

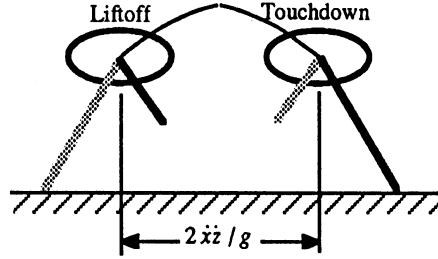


Figure 6: The distance traveled during flight is determined by the liftoff velocity $(\dot{x}, \dot{z})$. For symmetric steps, in which the height of the body above the ground is the same at touchdown as at liftoff, the duration of flight is $2\dot{z}t_o/g$. The forward progress during flight is $2\dot{x}\dot{z}/g$. The system in the figure is moving from left to right, lifting off from the foot behind the hip and landing on the foot in front of the hip.

formula to take the leg spring mechanical steps into account gives an expression for the stance duration:

$$T_s = \begin{cases} (2/\omega_0) [\pi - \arctan(N'_G)], & \text{if } N'_G > 0 \quad (P < mg); \\ \frac{\pi}{\omega_0}, & \text{if } N'_G = \infty \quad (P = mg); \\ (2/\omega_0) \arctan(N'_G), & \text{if } N'_G < 0 \quad (P > mg). \end{cases} \quad (6)$$

where

- N'_G is a modified Groucho Number ($N'_G = \frac{\dot{z}\omega_0/g}{P/mg-1}$),
- ω_0 is the natural frequency of the spring-mass system ($\sqrt{k/m}$),
- $\dot{z}$ is the vertical velocity at liftoff,
- g is the acceleration of gravity
- k is the leg stiffness,
- m is the body mass,
- P is the preload force.

The entry for stance duration in figure 1 is the third case of equation 6, where the preload force exceeds the weight of the system.

Equation 6 gives the stance duration of a system bouncing in place on a linear spring. Increasing the leg stiffness, the preload force, or the vertical velocity shortens the stance duration, whereas increasing the mass or the acceleration of gravity lengthens the stance duration. The behavior is qualitatively similar for the more complex case of a nonlinear spring and the leg pivoting about the foot. In the nonlinear case, equation 6 can be used to predict stance duration by assuming, computing, or measuring a value for the natural frequency. The natural frequency depends on the effective vertical stiffness, which depends on the sweep angle and impact velocity as well as on the stiffness of the leg.

3.3 Flight

The two nodes in the middle of row three of figure 1 describe the flight travel and the flight duration. The formulas give the duration of flight and forward progress of a rigid body that has an initial velocity $(\dot{x}, \dot{z})$ and is accelerated only by gravity (g). The formulas thus ignore changes in the location of the center of gravity due to the motions of the legs, and accelerations due to aerodynamic drag. The rigid body assumption leads to simple expressions for the flight duration and the flight travel:

$$T_f = 2\dot{z}/g \quad (7)$$

$$S_f = 2\dot{z}\dot{x}/g. \quad (8)$$

The third row of figure 1 divides running speed into four quantities: stance travel, stance duration, flight travel, and flight duration. These four quantities depend on the quantities in row four, which are functions of the operating point, and on the parameters in row five, which characterize the mechanism. The behavior in stance depends on the mass of the system, the stiffness and preload of the leg spring, the length and angle of the leg at liftoff, and the regulation of body attitude. The behavior in flight depends on the joint positions and velocities at liftoff. The following sections discuss the dependencies on operating point and mechanism parameters in more detail.

3.4 Stance Travel

The system travels a distance $S_s = 2\ell \sin(\theta)$ during stance. The motion during stance depends on the leg length and leg angle at touchdown, which are chosen by the control system to make the motion during stance symmetric and thus maintain a constant forward speed. The higher the speed, the farther forward the foot must be ahead of the hip at touchdown. The design of the leg limits the leg length, and the design of the hip joint limits the angle of the leg with respect to the body.

The leg angle is limited by the angle of the leg with respect to the body, and by the angle of the body with respect to the ground. We call the angle of the body with respect to the ground the pitch of the body. With the body in its nominal orientation, the range of leg angles permitted by the hip joint is symmetric about vertical, allowing the leg to swing equally far forward and backward. Deviations of the body from its nominal orientation reduce either the distance that the leg can reach forward for touchdown or the distance that it can reach back before liftoff. Either case reduces top speed by reducing the travel that can be achieved during stance.

If the body rotates away from its nominal pitch angle, the travel of the foot with respect to the hip is asymmetric. If the body is pitched forward, the distance that the foot can reach ahead at touchdown is reduced. During flight, the control system positions the foot in front of the hip by a distance that is proportional to running speed. Reducing the foot travel ahead of the hip reduces the maximum stable running speed, regardless of how much foot travel is gained behind the hip. If the body rotates backward, the foot can reach farther ahead of the hip for touchdown. However, the hip joint then reaches its limit of rearward travel before the end of stance, abruptly pitching the body forward.

3.5 Stance Duration

The duration of stance is $T_s = (2/\omega_0) \arctan\left(\frac{\dot{z}\omega_0/g}{P/mg-1}\right)$. The expression is based on the assumption that the stance duration is determined by the vertical motion of the system independent of the horizontal motion. Other plausible expressions for the stance duration might be derived from the strength of the leg, from the decrease in forward speed during stance, and from the velocity and acceleration limitations of the hip joints. For figure 1, we chose the representation based on stiffness, because leg stiffness *determines* stance duration, while leg strength and hip joint properties might *limit* stance duration. The stance duration must be long enough that the forces do not break the leg, and that the hip joint has time to move from the touchdown angle. The stance duration must be long enough, or the speed slow enough, that the hip joint does not exceed its maximum angle of rotation.

3.6 Flight Travel and Flight Duration

During flight the center of gravity of the system moves along a parabolic trajectory determined by the velocity at liftoff and the acceleration of gravity. The liftoff velocity depends on the leg length, leg extension rate, leg angle and leg rotation rate. It has magnitude $\sqrt{\ell^2\dot{\theta}^2 + \dot{\ell}^2}$ and direction $\arctan(\dot{\ell}/\ell\dot{\theta}) - \theta$. The magnitude is independent of the leg angle. The horizontal and vertical components are:

$$\dot{x} = \dot{\ell} \sin(\theta) + \ell \dot{\theta} \cos(\theta) \quad (9)$$

$$\dot{z} = \dot{\ell} \cos(\theta) - \ell \dot{\theta} \sin(\theta). \quad (10)$$

3.7 Speed Equations

Combining the formulas in figure 1 yields a single equation that expresses running speed as a function of operating point and mechanism parameters. The top row of the tree defines running speed as $V = S/T$. Breaking up the step length and step period expands the definition to:

$$V = \frac{S_f + S_s}{T_f + T_s}. \quad (11)$$

Incorporating the definitions of stance travel, flight travel, flight duration, and stance duration gives:

$$V = \frac{\dot{x}\dot{z} + g\ell \sin \theta}{\dot{z} + (g/\omega_0) \arctan \left(\frac{\dot{z}\omega_0/g}{P/mg - 1} \right)}. \quad (12)$$

Finally, replacing the Cartesian components of liftoff velocity according to equations 9 and 10, gives an equation for running speed in terms of the state variables $(\ell, \dot{\ell}, \theta, \dot{\theta})$ at liftoff and the parameters (ω_0, P, m, g) :

$$\begin{aligned} & (\ell, \dot{\ell}, \theta, \dot{\theta}, \omega_0, P, m, g) \quad (13) \\ &= \frac{(\dot{\ell}^2 - \ell^2 \dot{\theta}^2) \sin(2\theta) + 2\ell \dot{\ell} \dot{\theta} \cos(2\theta) + 2g\ell \sin(\theta)}{2\dot{\ell} \cos(\theta) - 2\ell \dot{\theta} \sin(\theta) + (2g/\omega_0) \arctan \left\{ \left(\frac{\omega_0/g}{P/mg - 1} \right) [\dot{\ell} \cos(\theta) - \ell \dot{\theta} \sin(\theta)] \right\}}. \end{aligned}$$

The extent of the state space is determined by mechanism parameters that describe maximum excursions and velocities of the joints. The construction of the leg establishes minimum and maximum leg lengths, and the rate of change of leg length is less than some maximum. If the pitch angle is always zero, then the leg angle and leg rotation rate are limited by the maximum excursion and velocity of the hip joint. At liftoff the leg angle, leg rotation rate, and leg extension rate are all positive:

$$\begin{aligned} \ell_{\min} &\leq \ell \leq \ell_{\max} \\ 0 &\leq \dot{\ell} \leq \dot{\ell}_{\max} \\ 0 &\leq \theta \leq \varphi_{\max} < \pi/2 \\ 0 &\leq \dot{\theta} \leq \dot{\varphi}_{\max}. \end{aligned} \quad (14)$$

The operating point, which is the state at liftoff, lies in this restricted region of the state space.

3.8 Summary of the Dependency Tree

The top of the dependency tree shown in figure 1 is a performance measure, running speed. The intermediate rows are characteristics of the running motion:

- step rate
- step length
- flight duration
- stance duration
- stance travel
- flight travel
- leg sweep angle
- horizontal and vertical velocity at liftoff
- natural frequency

At the bottom of the tree are the parameters of the physical mechanism that determine running speed:

- body attitude
- leg length
- hip position
- leg extension rate
- hip rotation rate
- leg spring preload force
- leg stiffness
- mass

The formulas in the dependency tree are based on the following assumptions:

- There is no air drag, so the speed during flight is constant.
- The center of gravity is fixed with respect to the body, so the mechanism moves like a rigid body during flight.
- The motion during stance is symmetric:

$$\begin{array}{llll}
 \ell_{td} = \ell_{lo} & \theta_{td} = -\theta_{lo} & x_{td} = -x_{lo} & z_{td} = z_{lo} \\
 \dot{\ell}_{td} = -\dot{\ell}_{lo} & \dot{\theta}_{td} = \dot{\theta}_{lo} & \dot{x}_{td} = \dot{x}_{lo} & \dot{z}_{td} = -\dot{z}_{lo}
 \end{array}$$

This structure provides a framework for studying how running speed depends on the operating point and on the physical mechanism parameters.

The dependency tree provides a model of running speed. Although it incorporates several simplifying assumptions, it provides more intuition about how running speed depends on mechanism parameters than more complex models would. In particular, dynamic simulations would predict the results of experiments, but they have too many parameters to offer much insight to the problem. The dependency tree models running with a small number of parameters, and shows qualitatively how each affects speed.

4 Experiments

We have experimented with the planar biped to study how the design and control of a legged system affect its top running speed. The planar biped runs faster with long legs than with short legs, and faster with stiff leg springs than with soft leg springs. Experimenting with the biped has made it clear that there are speed dependent disturbances to body attitude, and that fast running requires that the control system reject or correct those disturbances. The biped's power dissipation increases with running speed, but the increase is small compared with the power required just to run in place.

4.1 How Fast Running Differs from Slow Running

Figure 7 shows the motions of the legs and body as the biped ran forward at a constant speeds of 1, 3, and 5 m/s, respectively. Table 3 lists the properties of a typical step at each speed. Compared to running slowly, running fast was characterized by longer and more frequent steps, higher frequency oscillations of leg length, leg angle, and body attitude, smaller and more frequent vertical oscillations of the body, and larger angular motions of the legs.

5 Leg Length

The longer a legged system's legs are, the faster it can run. Figure 8 shows the results of nine experiments with the planar biped, each with a different leg length. During flight, the control system servoed the active leg to the indicated length. During stance, the leg actuator extended, so the leg was longer at liftoff than it was at touchdown. During each experimental run, I used a joystick to increase the desired running speed, attempting to find the highest speed at which the biped would run without losing its balance. The reported speed for each run is the highest average speed for one lap of the 16m running track.

The control system could select leg lengths between 0.50m and 0.65 m by adjusting the leg actuator. For longer leg lengths, the biped's legs were

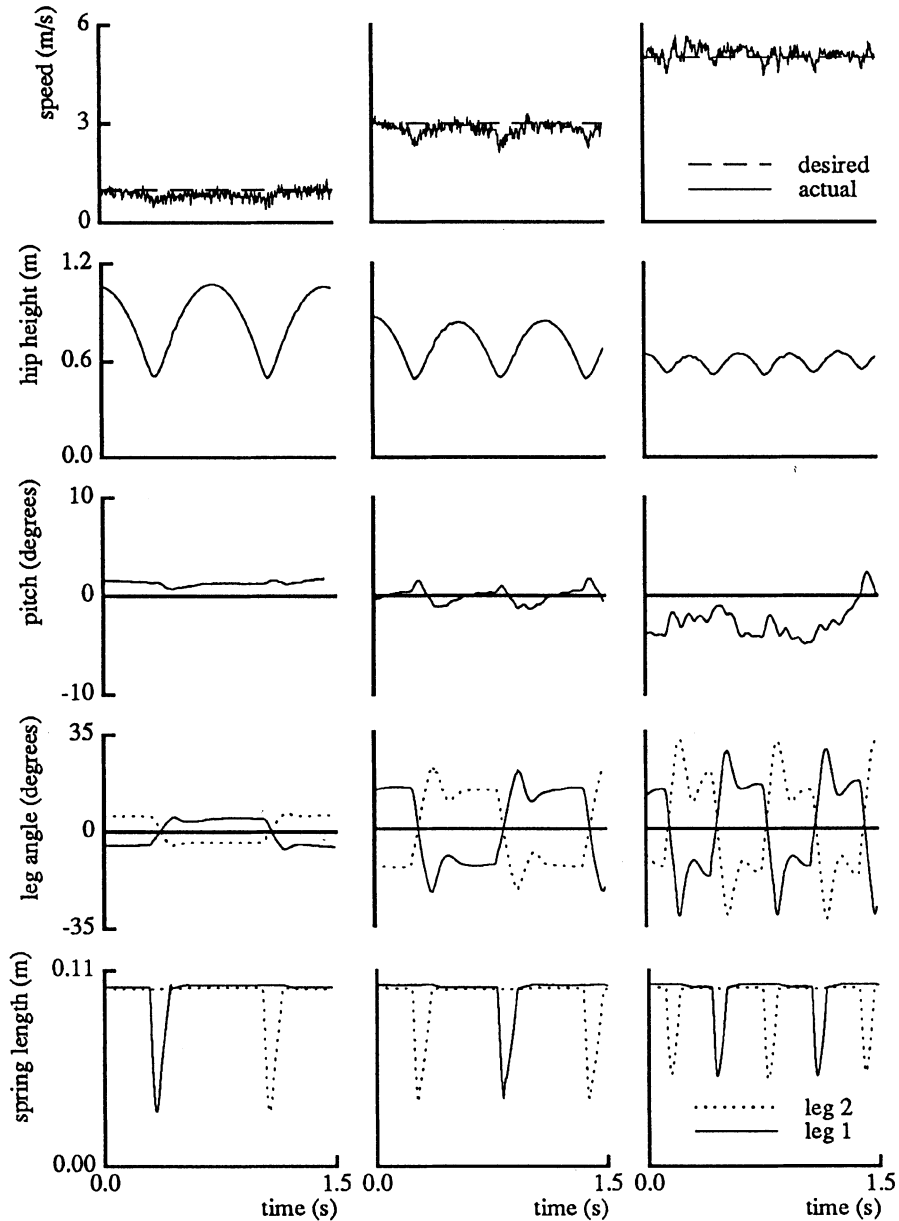


Figure 7: The planar biped ran forward at 1, 3, and 5 m/s. The top three graphs show the forward speed, hip height, and pitch angle of the body. The fourth graph shows the equal and opposite motions of the two legs as they sweep back and forth. The bottom graph shows how the two leg springs compressed as the biped bounced alternately on the two feet. As running speed increased, the frequency of stepping and leg angle excursion increased, and the compression of the legs during the bounce decreased.

Step Parameters at 1, 3, 5 m/s

commanded speed (m/s)	1.0	3.0	5.0
observed speed (m/s)	0.90	2.92	4.99
stance duration (s)	0.128	0.128	0.088
flight duration (s)	0.588	0.392	0.320
step period (s)	0.716	0.520	0.320
stance travel (m)	0.10	0.34	0.43
flight travel (m)	0.55	1.18	1.17
step length (m)	0.65	1.52	1.60
leg length (m)			
touchdown	0.595	0.595	0.591
liftoff	0.666	0.658	0.620
leg angle w.r.t vertical (°)			
touchdown	-2.2	-11.1	-19.5
liftoff	7.4	21.9	25.2
body angle w.r.t horizontal (°)			
touchdown	1.3	0.0	-4.1
liftoff	1.3	-0.2	-2.5
vertical velocity (m/s)			
touchdown	-2.95	-2.38	-1.60
liftoff	3.00	2.12	1.54
horizontal velocity (m/s)			
touchdown	0.75	2.58	5.02
liftoff	0.86	2.97	4.97

Table 3: Each column of data is for a typical step on leg two of the planar biped while it was running at constant speed. Increasing speed was accompanied by longer steps and shorter step periods, as shown by these data. During these experiments, the air pressure in the leg springs was 90 psi and the thrust algorithm extended the leg actuator as quickly as possible during stance.

extended with stilts and joined to the bottom of the legs. The feet were moved to the bottom of the stilts. A 0.191 m stilt gave a leg length of 0.844 m, and a 0.391 m stilt gave a leg length of 1.005 m. Figure 8 shows that the top running speed of the biped increased with increasing leg length, but may flatten out above 0.8 m.

The increasing leg length of the biped was not accompanied by other changes specified by any principle of geometric or elastic similarity. No dimension other than leg length changed. Since the diameter of the legs did not change, the strength of the legs remained the same, and the factor of safety against structural failure got smaller as the legs got longer. Figure 9 shows one of the consequences. After several running experiments at the longest leg length, one of the stilts broke where it was attached to the leg. The bending force on the leg had torn the stilt where it was fastened to the leg.

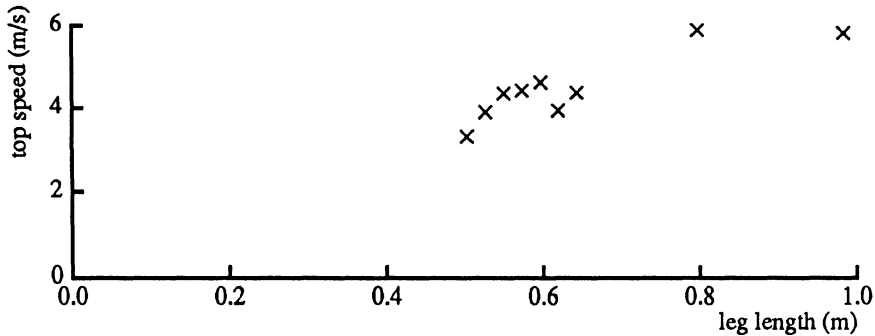


Figure 8: The longer the legs, the faster the planar biped ran. The planar biped ran nine times, each time with a different leg length. During flight, the control system adjusted the length of the leg. In each run, the experimenter raised the forward running speed to the highest value that could be maintained without the biped losing its balance. The listed speed is the average for a complete circuit around the 16 m circular track. Initially the range of possible leg lengths was 0.50 m to 0.65 m. Longer leg lengths were obtained by adding stilts to the end of the biped's legs. A 0.191 m stilt gave a leg length of 0.844 m, and a 0.391 m stilt gave a leg length of 1.005 m. During the experiments with the stilts, the leg actuator extended as fast as possible during stance, rather than trying to extend 0.021 m as in the other experiments. With a leg length of 0.844 m, the biped ran 5.9 m/s (13.1 mph), the highest speed ever recorded.

The leg springs did not get any longer when the legs got longer, and that may have caused a problem. If the body were to move horizontally during stance, then the leg length when the leg was vertical would be $\ell \cos(\theta)$, where ℓ and θ are the leg length and leg angle at touchdown. So when the leg was vertical, the leg spring would be deflected at least $\ell[1 - \cos(\theta)]$ from its length at touchdown. The actual deflection of the spring would be greater, since the path of the body is concave upward, so that the hip is always lower in the middle of stance than at touchdown or liftoff. The air springs on the biped have a maximum deflection of less than 0.10 m. For a leg angle of 25° , a leg length of 0.66 m would require the spring to deflect at least 0.06 m, but for a leg length of 0.98 m, the required deflection would be more than 0.09 m. The limit on leg spring deflection probably prevented the biped from using its full hip travel when it was running with the long stilts.

When the long stilts were on the machine, the hip servo position gain had to be reduced to keep the servo from oscillating. Lengthening the legs increased their moment of inertia and their flexibility, both of which lowered the natural frequency of the first mode of vibration of the leg. Lowering the position gain softened the servo so that it did not excite the vibration of the leg.

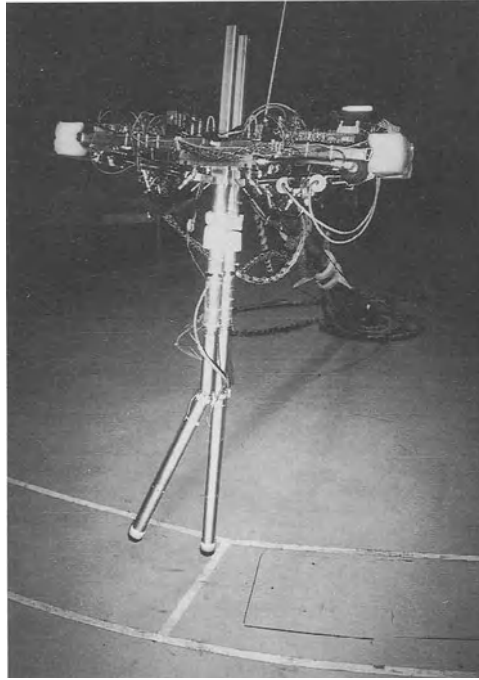


Figure 9: One of the long stilts broke after several running experiments. The wall of the tubing tore where it was screwed onto the plug that joined the stilt to the leg tube. Long legs are more vulnerable to buckling failure than short legs.

The planar biped runs faster with long legs than with short legs. The broken stilt and the need to soften the hip servo point out some of the problems that accompany increasing leg length.

6 Leg Stiffness

The stiffer the leg springs are, the faster a legged system can run. Figure 10 shows the results of eight experimental runs with the planar biped, each at a different leg stiffness. Before each experimental run, I set the indicated pressure with the regulator that supplies air to the pneumatic leg springs.

During the runs, I used a joystick to gradually increase the desired running speed to find the highest speed at which the biped would run without losing its balance. The reported speed for each run is the highest average speed for one lap of the 16 m circumference running track. The reported stance duration is the average of the stance durations observed during the fastest lap.

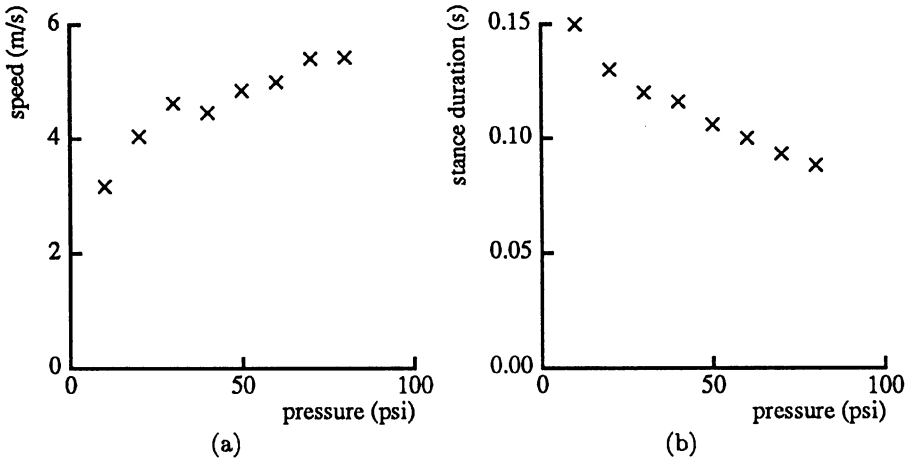


Figure 10: These plots show the variation of top running speed (a) and of stance duration (b) as a function of the air pressure in the leg springs. Each point represents the average stance duration or forward speed over the fastest lap at the given air pressure. To generate the data the planar biped ran eight times, each with a different air pressure in the leg springs. In each run, the experimenter raised the forward running speed to the highest value that could be maintained without the biped losing its balance. The reported forward speed, stance duration, and vertical landing velocity are average values for a complete circuit around the 16 m circular running track. During these runs, the leg length at touchdown was 0.623 m, and the thrust algorithm extended the leg actuator as rapidly as possible during stance.

The leg stiffness of the planar biped depends on how much air is in the leg springs. An air line leads from the spring chamber to a regulator that maintains the desired pressure in the line. A check valve isolates the spring chamber when pressure inside is higher than the pressure in the line. If air leaks out of the spring while it is compressed, then when the spring extends the pressure inside drops below the pressure in the air line. In this case the check valve opens, restoring the spring pressure to the desired value. The higher the air pressure, the more air there is inside of the leg spring. Higher air pressure increases both the stiffness and the preload force of the spring.

Figure 10 shows that the biped ran faster and took steps with shorter stance duration when it was running with high leg spring air pressure than when

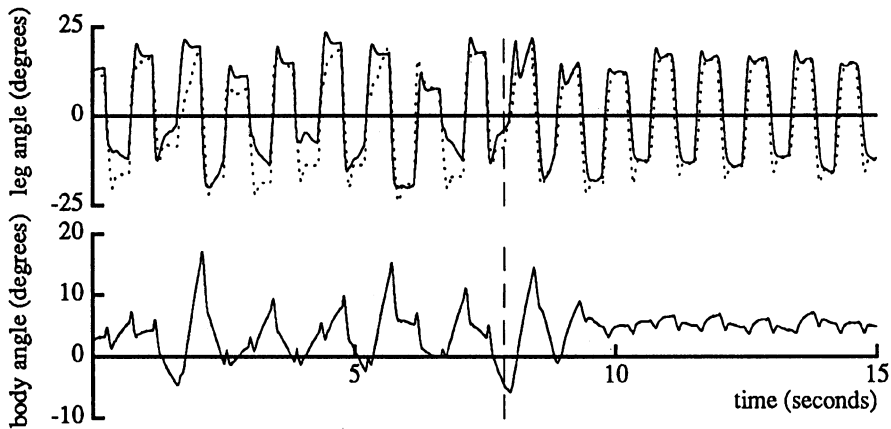


Figure 11: A control algorithm that kept the leg angles equal and opposite reduced the the amplitude of the oscillations in body attitude. The top graph shows the angle of each leg with respect to the axis of symmetry of the body. The sign of the angle of leg 2 is reversed so that when the leg angles are symmetric the lines are on top of one another. The vertical line marks a switch from an algorithm that moved the legs independently to one that ensured that the leg angles were mirror images. The axis of symmetry was a line passing through the hip joint perpendicular to the body. The bottom graph shows that changing the leg positioning algorithm reduced the oscillations in body angle from about 20° peak-to-peak to about 6° peak-to-peak. In this experiment the planar biped was running about 2.5 m/s (5.6 mph).

it was running with low leg spring air pressure. The stiffness of the legs, and thus the natural frequency of the bouncing motion increased with the pressure.

7 Body Attitude

Body attitude is important to top running speed because of the limited range of motion of the hip joints. If the body tips forward during flight, the hip joint limit prevents the foot from reaching as far forward for landing as it can with a level body. The distance that the foot needs to reach forward for landing is proportional to running speed, so forward tipping of the body reduces the top running speed. If the body tips backward during stance, the hip joint limit prevents the foot from reaching as far backward as it can with a level body. In this case, the hip may reach the joint limit before the leg leaves the ground, causing a sudden forward pitching of the body. Top running speed requires good control of body attitude so the hip joint can sweep through its full range of motion during stance. If there were no kinematic limits to hip angle, then body attitude would not affect running speed.

7.1 Mirroring

Hip torques that position the legs also rotate the body. The control system can minimize disturbances to the body attitude by ensuring that the two legs move at the same time and in opposite directions. During stance, while the support leg sweeps backward, the other leg swings forward. During flight, while one is positioned for landing, the other leg makes compensating motions to reduce the torques on the body. This mirroring action substantially reduces the variation in body attitude that occurs when each leg is moved independently.

Figure 11 shows the result of an experiment comparing two different algorithms for moving the idle leg. At the time marked by the vertical dotted line, the control system switched algorithms. Before that time, the legs were positioned independently, and afterwards they were positioned according to the mirroring algorithm. The mirroring algorithm reduced the oscillations in pitch angle, from about 20° peak-to-peak to about 6° peak-to-peak.

When the two legs were being positioned independently, the algorithm was as follows: during stance, the stance leg was swept back by the forward motion of the body, and by hip torques selected by the body attitude control servo. A leg angle servo moved the swing leg forward into position for the next touchdown. During flight, leg angle servos positioned the leg that would touch down next for landing, and servoed the leg that had just lifted off to the angle it had at liftoff. The leg angle servos were as stiff as possible, in order to minimize steady state error. The swing leg moved forward very quickly and stopped at the desired position well before the end of stance. After liftoff, the leg that had just left the ground was still rotating, so the servo applied torques to stop the rotation and move the leg back to the position it had at liftoff. These torques disturbed the body attitude.

The mirroring algorithm servoed the idle leg so that its hip angle was equal and opposite to that of the active leg. During stance, the support leg was active, and swept back as the body moved forward. The swing leg was idle, and moved forward at the same rate as the support leg moved back. During flight, the leg that would next touch down was active, and was positioned for landing as usual. The leg that had just lifted off was idle, and made motions symmetric to the active leg. Compared with the independent positioning algorithm, mirroring caused the swing leg to advance more slowly, which reduced the disturbance to body attitude. During flight, mirroring caused the reaction torques generated by positioning one leg to be canceled by torques from moving the other leg.

7.2 Sweep Compensation

The hip actuators on the planar biped have internal damping that causes a velocity dependent discrepancy between the commanded force and the delivered force. When the biped runs fast, it leans forward until the body attitude servo commands enough force to overcome the actuator damping. Explicitly compensating for the velocity dependent forces reduces the tendency to lean forward.

Figure 12 shows the body angle during an experimental run in which the control system gradually increased the running speed from zero to 4 m/s. The graphs show that the average body angle increased as running speed increased.

When the biped runs in place, the hips barely move. When it runs forward, each hip rotates one way as the leg sweeps back during stance and the other way as the leg swings forward in preparation for the next step. These hip joint velocities are proportional to running speed. Each biped hip actuator behaves like a torque source in parallel with a damper. For a given input signal, the actuators produce less force when they are moving quickly than when they are moving slowly. The damping forces are proportional to hip rotation rate, which is proportional to running speed.

The hip torque to overcome the actuator damping comes from the body attitude servo. During stance, the servo applies torques proportional to the angle and angular velocity of the body. The angle stabilizes when the body has leaned forward enough that the attitude servo generates a correcting torque equal to the torque caused by the actuator damping force. Because the force is proportional to running speed, the body leans farther forward as the biped runs faster.

Compensating for the velocity dependent torques reduces the lean of the body. Figure 13 shows the body angle during an experimental run in which the body attitude servo added negative damping to the hip actuators by feeding back a signal proportional to the actuator velocity. The signal to the servovalve was:

$$\tau = k_{\rho}\rho + k_{\dot{\rho}}\dot{\rho} + k_{\dot{w}}\dot{w}, \quad (15)$$

where ρ and $\dot{\rho}$ are the pitch angle and the pitch rate, $\dot{w}$ is the hip actuator velocity, and k_{ρ} , and $k_{\dot{\rho}}$ are the position and velocity gains that control pitch, and $k_{\dot{w}}$ is the inverse damping coefficient. The modification reduced the body angle offset from about 7° to about 3° . The same result might have been obtained by adding an integral term to the body attitude servo.

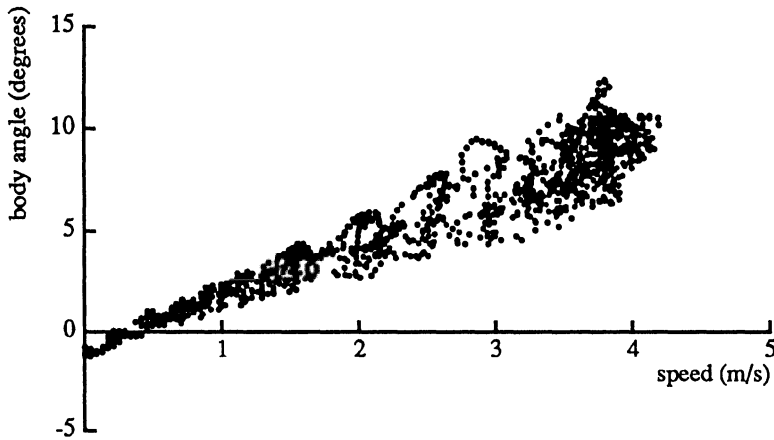


Figure 12: The planar biped leans forward when it runs fast. The graph shows body angle plotted as a function of running speed. The body angle oscillated during each step. As the running speed increased, the peak-to-peak amplitude increased with a slope of about $2.0^{\circ}/\text{m}$. Similarly, the average body angle increased with a slope of about $2.2^{\circ}/\text{m}$. At 4 m/s the offset was 7° and the peak-to-peak amplitude was 5° .

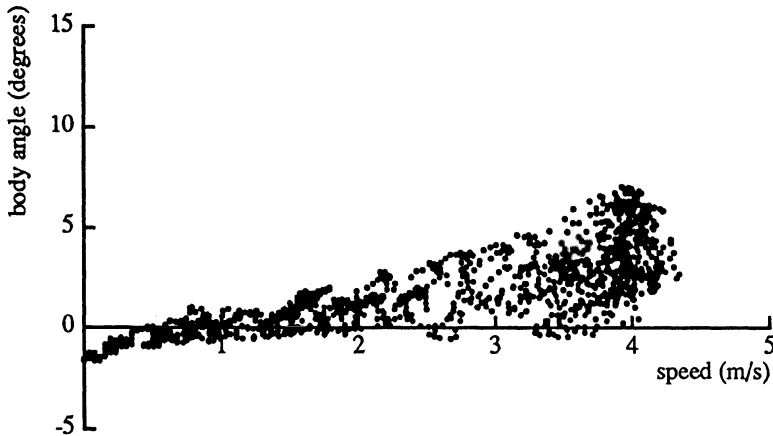


Figure 13: Compensating for velocity dependent forces in the hip actuators reduced the average body angle, but did not reduce the amplitude of oscillation. As in figure 12, the body angle took on an offset and an oscillation as the running speed increased. Sweep compensation reduced the average body angle but not the amplitude of the oscillations. The increase in offset was about $1.5^{\circ}/\text{m}$. At 4 m/s the offset was about 3° and the peak-to-peak amplitude was about 6° .

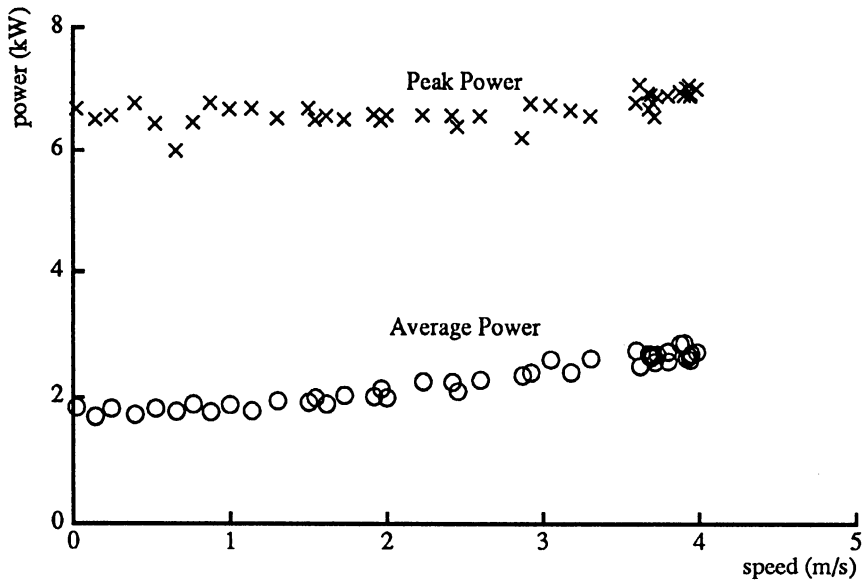


Figure 14: The planar biped dissipates slightly more energy to run fast than it does to run slowly. The top two graphs show the instantaneous power and the running speed as the biped accelerated from rest to 4 m/s. The bottom graph shows power plotted as a function of running speed. The crosses show the peak instantaneous power on each step, and the circles show the average power for each step. The peak power was a nearly constant 6.6 kW, increasing very slightly at speeds of about 4 m/s. The average power increased gradually, from 1.8 kW for hopping in place, to 2.7 kW for running 4 m/s.

8 Power Dissipation

The planar biped dissipates slightly more power when it runs fast than when it runs slowly. Figure 14 shows the power dissipated during an experimental run in which the control system gradually increased the running speed from zero to 4 m/s. The peak power was nearly constant, increasing very slightly at 4 m/s. The average power increased gradually as running speed increased. The instantaneous power shown is the product of the supply pressure measured by a sensor on the robot, and the total flow computed from the actuator velocities. The flow computation included an estimate of the flow that leaked through the servovalves, which does not appear in the actuator velocities. The estimated leakage was about 39 cc/s, which at a system pressure of 3000 psi corresponds to 0.8 kW. The hydraulic pump maintained a nearly constant pressure, so the power was proportional to the flow.

A 7.5 kW motor drives the hydraulic pump. The peak instantaneous power of 6.6 kW probably exceeds what the pump can deliver. However, the 5 gal hydraulic accumulator, the compliance of the hydraulic hoses, and the inertia

of the oil all filter out flow transients, so the pump never has to deliver the peak power. The average power dissipation was less than 3 kW, which is well within the capacity of the pump and motor. The biped's running speed is not currently limited by the available power.

We measured the power dissipation during a pair of experiments in which the planar biped ran using two different gaits. In the first experiment, the biped ran with its usual alternating two-legged gait. In the second experiment, it ran by hopping on one leg. The second leg stayed short, and moved back and forth to compensate for the reaction torques of the active leg.

The biped ran at about the same speed with both gaits, but dissipated less energy when it ran on only one leg. It ran 5.3 m/s on two legs, and dissipated 3.5 kW, with instantaneous peaks of 7.2 kW. On one leg, it ran 5.4 m/s, dissipated 2.9 kW, with instantaneous peaks of 6.0 kW.

Running on one leg required the legs to sweep back and forth twice as frequently as they did in two-legged running, so the hip actuators dissipated more power. On the other hand, running on one leg meant that the other leg never had to change length. The leg actuator has a large area and a long stroke, so moving it causes a large flow that dissipates a lot of power without doing any work. Keeping one leg short and not moving its actuator saved more than enough energy to compensate for the increased dissipation of the hip actuators.

The biped's hydraulic system is very inefficient for applying small forces at high velocities. The pump supplies oil at constant pressure, so the power supplied is proportional to the flow of oil. To apply a small force, an actuator throttles the oil down to a lower pressure, dissipating energy. That the biped dissipated less power running on one leg than running on two is an artifact of the constant pressure hydraulic system.

9 Summary

The running speed of a legged system depends upon the frequency and length of its steps. The time required for a step can be reduced by stiffening the legs, and the step length can be increased by lengthening the legs. If body attitude is not well controlled, the limited range of motion of the hips limits the length of the steps.

Experiments with the planar biped showed that it runs faster with stiff legs than with soft legs, and that it runs faster with long legs than with short legs. To get it to run fast, the control system reduces variation of body attitude by moving the legs symmetrically, and by compensating for velocity dependent hip actuator forces. The biped's power dissipation is well within the capacity of its power supply.

During its fastest run the planar biped ran 5.9 m/s (13.1 mph) on long, stiff legs. The leg length at landing was 0.844 m, and the air spring pressure was 85 psi. The control system moved the legs symmetrically, and compensated for hip actuator damping forces.

Acknowledgements

The authors would like to thank Jessica Hodgins for her help with this research, and Ben Brown and Jeff Miller for their help in designing and building the planar biped. This research has been supported by funds from the Defense Advanced Research Projects Agency, System Development Foundation, and the National Science Foundation.

References

- R. McN. Alexander, V. A. Langman, and A. S. Jayes. 1977. Fast locomotion of some African ungulates. *J. Zoology, (London)* 183:291-300.
- R. McN. Alexander and A. Vernon. 1975. The mechanics of hopping by kangaroos (Macropodidae). *J. Zoology (London)* 177:265-303.
- R. McN. Alexander. 1988. *Elastic Mechanisms in Animal Movement*. Cambridge: Cambridge University Press.
- T. J. Dawson and C. R. Taylor. 1973. Energetic cost of locomotion in kangaroos. *Nature* 246:313-314.
- J. Furusho, M. Masubuchi. 1987. Control of a dynamical biped locomotion system for steady walking. In H. Miura, I. Shimoyama, editors, *Study on Mechanisms and Control of Bipedes*, Tokyo: University of Tokyo, 116-127.
- G. Gabrielli and T. H. von Kármán. 1950. What price speed? *Mechanical Engineering* 72:775-781.
- V. S. Gurfinkel, E. V. Gurfinkel, A. Yu. Shneider, E. A. Devjanin, A. V. Lensky, L. G. Shitilman. 1981. Walking robot with supervisory control. *Mechanism and Machine Theory* 16:31-36.
- A. V. Hill. 1950. The dimensions of animals and their muscular dynamics. *Science Progress* 38:209-230.
- S. Hirose. 1984. A study of design and control of a quadruped walking vehicle. *International J. Robotics Research* 3:113-133.
- D. F. Hoyt and C. R. Taylor. 1981. Gait and the energetics of locomotion in horses. *Nature* 292:239-240.
- M. Huang and K. J. Waldron. 1987. Relationship between payload and speed in legged locomotion. In *Proceedings of Conference on Robotics and Automation*, IEEE, Raleigh, NC. 533-538.
- T. Kato, A. Takanishi, H. Jishikawa, I. Kato. 1983. The realization of the quasi-dynamic walking by the biped walking machine. In A. Morecki, G. Bianchi, K. Kedzior, editors, *Theory and Practice of Robots and Manipulators, Proceedings of RoManSy'81*, Warsaw: Polish Scientific Publishers. 341-351.

- T. A. McMahon. 1975. Using body size to understand the structural design of animals: quadrupedal locomotion. *J. Appl. Physiol.* 39:619–627.
- T. A. McMahon and P. R. Greene. 1978. Fast running tracks. *Scientific American* 239:148–163.
- T. A. McMahon and P. R. Greene. 1979. The influence of track compliance on running. *J. Biomechanics* 12:893–904.
- T. A. McMahon. 1985. The role of compliance in mammalian running gaits. *J. Exp. Biol.* 115:263–282.
- T. A. McMahon, G. Valiant, and E. C. Frederick. 1986. Groucho running. *J. Appl. Physiol.* 62:2326–2337.
- H. Miura, I. Shimoyama. 1984. Dynamic walk of a biped. *International J. Robotics Research* 3:60–74.
- M. H. Raibert. 1986. *Legged Robots That Balance*. MIT Press.
- I. E. Sutherland, M. K. Ullner. 1984. Footprints in the asphalt. *International J. Robotics Research* 3:29–36.
- M. Russel. 1983. ODEX I: The first functionoid. *Robotic Age*. 5(5):12–18.
- C. R. Taylor, N. C. Heglund, and G. M. O. Maloiy. 1982. Energetics and mechanics of terrestrial locomotion: I. Metabolic energy consumption as a function of speed and body size in birds and mammals. *J. exp Biol.* 97:1–21.
- K. J. Waldron, V. J. Vohnout, A. Pery, and R. B. McGhee. 1984. Configuration design of the adaptive suspension vehicle. *International J. Robotics Research* 3:37–48.

Robot Biped Walking Stabilized with Trunk Motion

Atsuo Takanishi

Department of Mechanical Engineering, Waseda University, Ookubo
Shinjuku-ku, Tokyo, Japan

Abstract

When walking in different environments, a biped walking robot must vary its gait (walking period and/or step length, etc) according to the environment. In order to realize that, I devised a universal control method for dynamic biped walking on a disturbance-free flat floor stabilized with trunk motion for a biped walking robot which has a trunk that serves as a balancing aid. The control method consists of two main parts. One is an algorithm that computes balancing motion of the trunk automatically from motion of the lower-limbs and a trajectory of the **ZMP**(Zero Moment Point) planned arbitrarily before the robot begins walking. The other is a program control of the walking using preset walking patterns transformed from motion of the lower-limbs and the trunk. In 1986, in order to confirm the effect of the control method, my coworkers and I developed the biped walking robot **WL-12**(Waseda Leg - 12), which has a trunk, and applied the control method to it. The **WL-12** realized several different gaits during dynamic walking that was stabilized by its trunk motion on a disturbance-free flat floor. The minimum time spent walking during these experiments was 1.3[sec] a step and the maximum step length was 0.3[m]. After that, we refined the **WL-12** and renamed it **WL-12R**(Refined). In 1988, the **WL-12R** succeeded in achieving faster walking of 0.8[sec] a 0.3[m] step.

A small modification of the control method effectively controls any kind of external forces and moments, which can be considered a kind of disturbance to a robot in walking, if they are known before the robot begins walking. We aimed at realizing dynamic biped walking that is stabilized with trunk motion under known external force by applying this modified method to the biped walking robot **WL-12R**. We developed a system that uses a **DD**(Direct Drive) motor to generate the external force that affects the robot when it is walking. As a result of experiments, the **WL-12R** realized various gaits under known external force in dynamic walking stabilized with trunk motion on a flat floor. The maximum force strength was 10[kgf].

1 Introduction

In 1984, my coworkers and I succeeded in attaining dynamic walking on a flat floor using a biped walking robot, the **WL-10RD**(Waseda Leg - 10 Refined Dynamic)[1]. The walking lasted 1.3[sec/step] with a 0.4[m] step. Moreover, in 1985, we accomplished the ascending and descending of stairs and slight inclines using the same robot.

But, in case of walking in different environments, I believe that a biped walking robot must vary its gait (walking period and/or step length, etc) according to the environment. From this point of view, a biped walking robot which consists of only the lower-limbs does not have the capability to change its gait.

Therefore, I devised a universal control method for dynamic biped walking on disturbance-free flat floor stabilized with trunk motion for a biped walking robot that has a trunk as a balancing aid. In 1986, in order to confirm the effect of the control method, my coworkers and I developed a biped walking robot, the **WL-12**(Waseda Leg - 12), which has a trunk, and applied the control method to the **WL-12**.

As a result of experiments, the **WL-12** realized various gaits in dynamic walking stabilized with trunk motion on a disturbance-free flat floor[2]. The minimum time spent walking was 1.3[sec] a step and the maximum step length was 0.3[m].

After the experiments, we improved the **WL-12** and named it **WL-12R**(Refined). In 1988, the **WL-12R** succeeded in achieving faster walking of 0.8[sec] a 0.3[m] step.

A small modification of the control method effectively controls any kind of external forces and moments, which can be considered a kind of disturbance to a robot in walking, if they are known before the robot begins walking. We aimed at realizing dynamic biped walking that is stabilized with trunk motion under known external force by applying this modified method to the biped walking robot **WL-12R**.

We developed a system that uses a **DD**(Direct Drive) motor to generate the external force that affects the robot when it is walking.

And as a result of experiments, the **WL-12R** realized various gaits under known external force in dynamic walking stabilized with trunk motion on a flat floor. The maximum force strength was 10[kgf].

In this paper, the author will introduce the control method on a disturbance-free flat floor, the development of the **WL-12** and the walking experiments in the first place, and secondly introduce the method for walking under known external forces and moments on a flat floor and the walking experiments using the **WL-12R**.

2 Control Method for Disturbance-Free Walking

The control method for walking on a disturbance-free floor consists of two main parts. One is an algorithm to compute balancing motion of the trunk automatically from motion of the lower-limbs and a trajectory of the **ZMP**(Zero Moment Point)[3] given arbitrarily before the robot begins walking. The other is a program control for walking using preset walking patterns[1] transformed from motion of the lower-limbs and the trunk.

In this section, the algorithm to compute balancing motion of the trunk is described.

2.1 Modeling of Robot and ZMP

Let a machine model be called a walking system from the point of view of system dynamics.

Let the walking system be defined as follows:

- 1) The walking system, including its trunk, is regarded as a model that has one particle m_0 for the trunk and n -particles m_i ($i=1, \dots, n$) for the lower-limbs as shown in Fig.1.
- 2) The floor for walking is a rigid horizontal plane that can not be moved by any strength of forces and moments(torques).
- 3) A Cartesian coordinate system **O-XYZ** is set, where the **Z** axis is vertical and the plane which is formed by the **X** and **Y** axes equals that of the plane of the floor. It is a fixed coordinate system.

The **ZMP** of walking motion of the system on the Cartesian coordinate system can be derived as follows:

At first, each vector is defined as shown in Fig.2. Then, we obtain an equation of motion at the arbitrary point **P**, which is obtained by applying **D'Alembert's Principle**[4].

$$\sum_i m_i (\mathbf{r}_i - \mathbf{P}) \times (\ddot{\mathbf{r}}_i + \mathbf{G}) + \mathbf{T} = \mathbf{0} \quad (1)$$

By modifying (1), the components of the **ZMP** can be given.

$$X_{zmp} = \frac{\sum_{i=0}^n m_i (\ddot{z}_i + g_z) x_i - \sum_{i=0}^n m_i (\ddot{x}_i + g_x) z_i}{\sum_{i=0}^n m_i (\ddot{z}_i + g_z)} \quad (2)$$

$$Y_{zmp} = \frac{\sum_{i=0}^n m_i (\ddot{z}_i + g_z) y_i - \sum_{i=0}^n m_i (\ddot{y}_i + g_y) z_i}{\sum_{i=0}^n m_i (\ddot{z}_i + g_z)}$$

A translationally moving coordinate $W-\overline{XYZ}$ is established on the waist of the robot in parallel with the fixed coordinate $O-XYZ$ to ease consideration of relative motion of the trunk's particle m_0 . $[X_q, Y_q, Z_q]$ are the coordinates of the origin of the moving coordinate $W-\overline{XYZ}$ relating to the origin of the $O-XYZ$.

Let (2) be modified into equations on the moving coordinates.

$$\begin{aligned}\bar{X}_{zmp} &= \frac{\sum_{i=0}^n m_i (\ddot{\bar{z}}_i + \ddot{z}_q + g_z) \bar{x}_i - \sum_{i=0}^n m_i (\ddot{\bar{x}}_i + \ddot{x}_q + g_x) (\bar{z}_i + z_q)}{\sum_{i=0}^n m_i (\ddot{\bar{z}}_i + \ddot{z}_q + g_z)} \\ \bar{Y}_{zmp} &= \frac{\sum_{i=0}^n m_i (\ddot{\bar{z}}_i + \ddot{z}_q + g_z) \bar{y}_i - \sum_{i=0}^n m_i (\ddot{\bar{y}}_i + \ddot{y}_q + g_y) (\bar{z}_i + z_q)}{\sum_{i=0}^n m_i (\ddot{\bar{z}}_i + \ddot{z}_q + g_z)}\end{aligned}\quad (3)$$

2.2 Solution for Trunk Motion

We modify (3) into equations in which terms about the trunk's particle are the left-hand side and the rest are the right-hand side, namely $\alpha(t)$ and $\beta(t)$.

$$\begin{aligned}\bar{z}_0 \ddot{x}_q + \bar{z}_0 g_x + \ddot{\bar{z}}_0 \bar{x}_{zmp} + (\bar{z}_0 + z_q) \ddot{\bar{x}}_0 - (\ddot{\bar{z}}_0 + \ddot{z}_q + g_z) \bar{x}_0 &= \alpha(t) \\ \bar{z}_0 \ddot{y}_q + \bar{z}_0 g_y + \ddot{\bar{z}}_0 \bar{y}_{zmp} + (\bar{z}_0 + z_q) \ddot{\bar{y}}_0 - (\ddot{\bar{z}}_0 + \ddot{z}_q + g_z) \bar{y}_0 &= \beta(t)\end{aligned}\quad (4)$$

In general, both equations of (4) interfere with each other and are non-linear differential equations. Because each equation has the same variable z_0 and the trunk is usually connected to the lower-limbs through rotational joints (so that z_0 is not linear to x_0 and y_0). Therefore, it is difficult to derive analytic solutions from these equations. So we assume that the trunk particle m_0 does not move vertically for the purpose of decoupling and linearizing them.

$$\ddot{\bar{z}}_0 = 0, \quad \bar{z}_0 + z_q = \text{constant}\quad (5)$$

Then, we obtain decoupled linear differential equations.

$$\begin{aligned}(\bar{z}_0 + z_q) \ddot{\bar{x}}_0 - g_z \bar{x}_0 &= \alpha'(t) \\ (\bar{z}_0 + z_q) \ddot{\bar{y}}_0 - g_z \bar{y}_0 &= \beta'(t) \\ (\alpha'(t) = \alpha(t) - \bar{z}_0 \ddot{x}_q - \bar{z}_0 g_x, \quad \beta'(t) = \beta(t) - \bar{z}_0 \ddot{y}_q - \bar{z}_0 g_y)\end{aligned}\quad (6)$$

As for the initial value problem, solved trajectories of equations (6) do not converge, because the roots of the characteristic equations of (6) are non-

negative real numbers. But, in case of steady walking, each particle of the lower-limbs and the ZMP movement are planned periodically for the moving coordinate $\mathbf{w} - \overline{\mathbf{XYZ}}$. In that case $\alpha(t)$ and $\beta(t)$ are known periodic functions, so that these equations have periodic solutions as particular solutions. In this case, each solution can be represented as a Fourier series. Therefore, by using FFT(Fast Fourier Transformation)[5], we can easily obtain approximate periodic solutions of (6) giving motion of the trunk's particle.

The approximate solutions derived above are effective only in case of slow speed walking or for a robot with a long trunk. In other cases, the ZMP produces considerable error. So I worked out a repeating algorithm to obtain strict periodic solutions of the nonlinear equations (4) as follows:

At first, we substitute the approximate periodic solutions of the linear equations (6) for the equations (3), and compute the trajectories of the ZMP from (3). We subtract errors between the computed trajectories of the ZMP and the planned time trajectories of the ZMP from terms of the ZMP in linear equations (6), and compute the approximate periodic solutions again. This operation is repeated until the error of the ZMP falls within tolerance levels. Therefore, we obtain strict periodic solutions of nonlinear equations (4) giving trunk particle motion whose precision of error is suitable for practical use. A flow chart of the repeating algorithm is shown in Fig.3.

As for the convergence of the solutions, I did not prove it, but checked it numerically using computer simulations when the parameters were changed. The example where the trunk length is a parameter that is changed is shown in Fig.4. Another example, where the walking time is the parameter changed, is shown in Fig.5. From these Fig.s, we can see that this method turns out to be useful.

2.3 Expansion into Complete Walking

This algorithm is applicable not only to steady walking but also to complete walking which starts from a static standing state and returns to a static standing state. That is, we regard the whole complete walking motion as one walking cycle, and apply the algorithm to it. Then, we can obtain trunk motion for the complete walking as shown in Fig.6.

It is necessary to leave a relatively long period of standing time before starting and after stopping walking. Because the image function (in frequency domain) of (6) is

$$\begin{aligned} \bar{x}_0(\omega) &= \frac{2a}{\omega^2 + a^2} \times b \\ \left(a &= \sqrt{\frac{gz}{\bar{z}_0 + z_q}} \quad b = \frac{-1}{2\sqrt{gz(\bar{z}_0 + z_q)}} \right) \end{aligned} \quad (7)$$

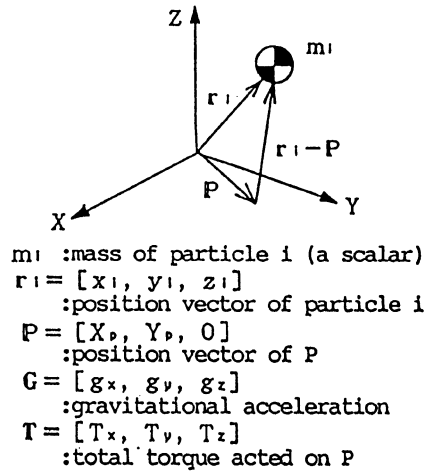
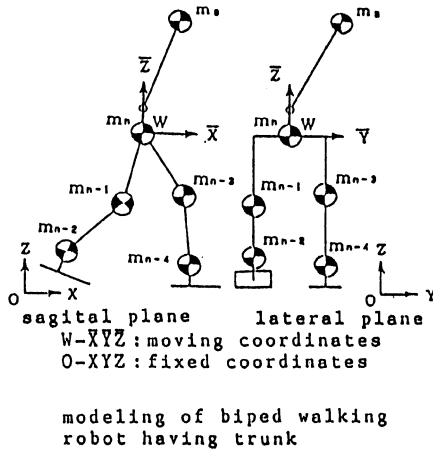


Fig.1 modeling of biped walking
 robot having trunk

Fig.2 definitions of vectors for
 walking system

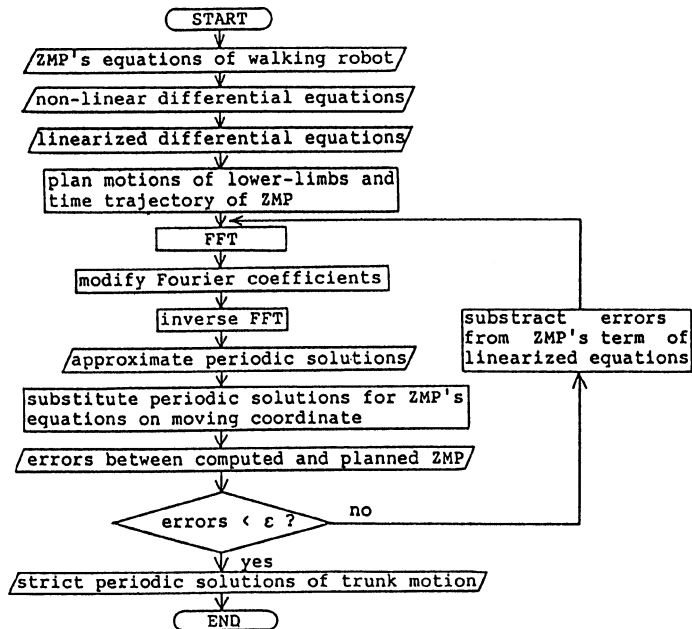


Fig.3 flow chart of iteration algorithm

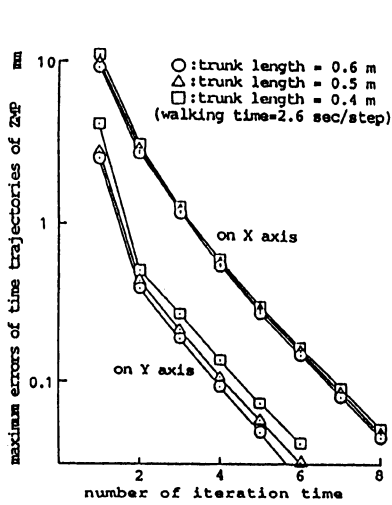


Fig.4 convergence of solutions
(parameter : trunk length)

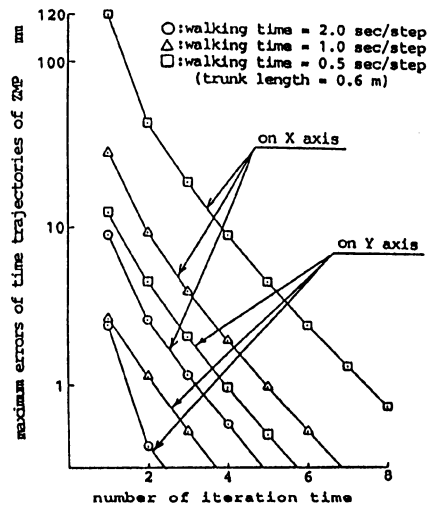
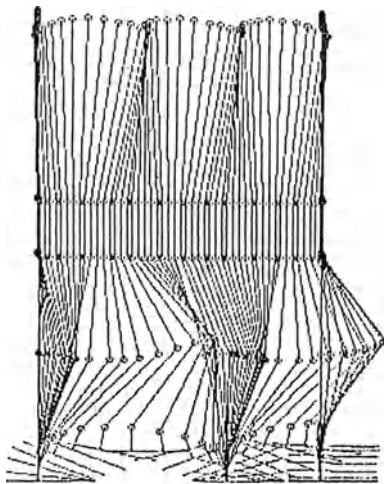


Fig.5 convergence of solutions
(parameter : walking time)



(sagittal plane)



(lateral plane)

Fig.6 example of complete walking computed by the iteration algorithm
(step : 0.3[m] , walking time : 1.3[sec/step])

So, the original function (in time domain) of (7) is

$$\bar{x}_0(t) = b \cdot e^{-a|t|} \quad (8)$$

From (8), we can see that it is necessary to take enough time for the velocity to be near to zero in the static standing state as shown in Fig.7.

3 Configuration of Biped Walking Robot WL-12

3.1 Machine Model WL-12

My coworkers and I developed a machine model, the **WL-12**, to test the control method. The **WL-12** weighed about 107[kg]. It stands about 1.8[m] tall when not walking. An assembly drawing and a photograph of the **WL-12** are shown in Fig.8 and 9.

The **WL-12** is made up of two legs as lower-limbs having 6 degrees of freedom(DOFs), and a trunk having 3 DOFs with a 30[kg] balancing mass as shown in Fig.10. The DOFs on each leg are rotational pitch axes on the hip, the knee and the ankle respectively. Two of the trunk's DOFs are rotational pitch and roll axes on the waist, and the remaining DOF is translational connecting the rotationals with the balancing mass.

An electro-hydro servo system, using an **RA**(Rotary Actuator) with a servo valve on each rotational joint and a hydraulic cylinder with a servo valve on the translational joint, are employed as actuators.

The robot's structural frame is made mainly of **CFRP**(Carbon Fiber Reinforced Plastic), and parts such as the actuators and manifold are made of duralumin, which reduced total weight.

As for sensors, the **RA** is equipped with a potentiometer and a tachometer generator that detects the rotational angle and angular velocity. The potentiometer and the tachometer generator are directly connected to the shaft of the **RA**, making feedback control correct and stable. Also, the **RA** is equipped with 2 pressure sensors in the hydraulic circuit between the **RA** and the servo valve. They monitor the output torque. The hydraulic cylinder is equipped with a linear potentiometer and a linear velocity sensor for feedback control. Each of the soles is equipped with 2 microswitches that monitor the floor contact of the toe and the heel.

3.2 Control System

A Control System is installed on the right and left sides of the waist separately, and controls the **WL-12** on a stand-alone basis.

The control system is shown as a block diagram in Fig.11. This system has a hierarchic structure, which consists of a main control board using a 16 bit CPU

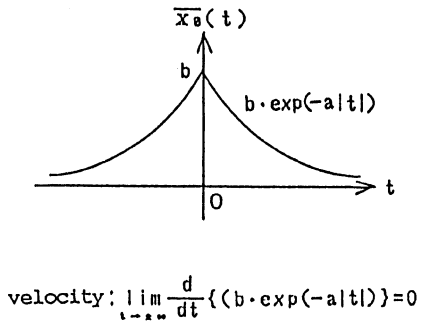


Fig.7 necessity of taking suitable time

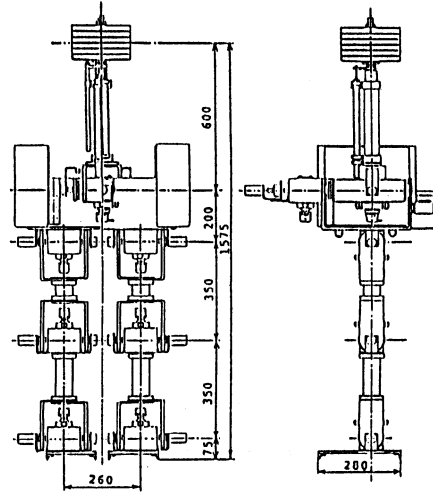


Fig.8 assembly drawing of WL-12

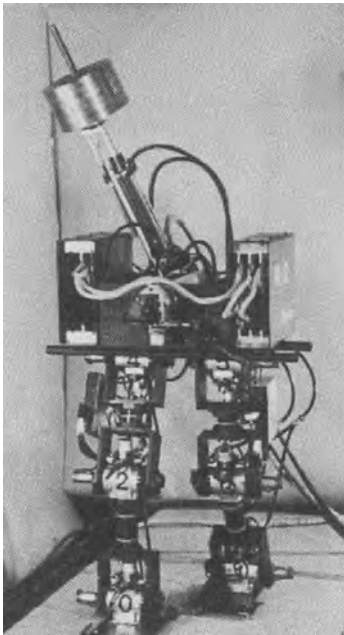


Fig.9 photo of WL-12

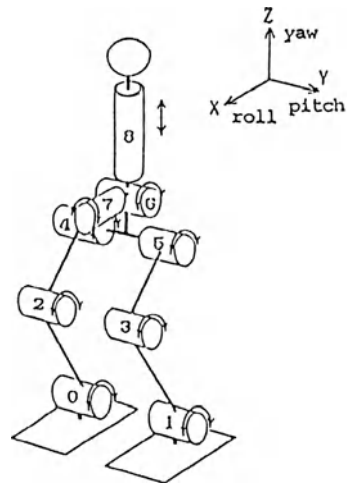


Fig.10 link structure of WL-12

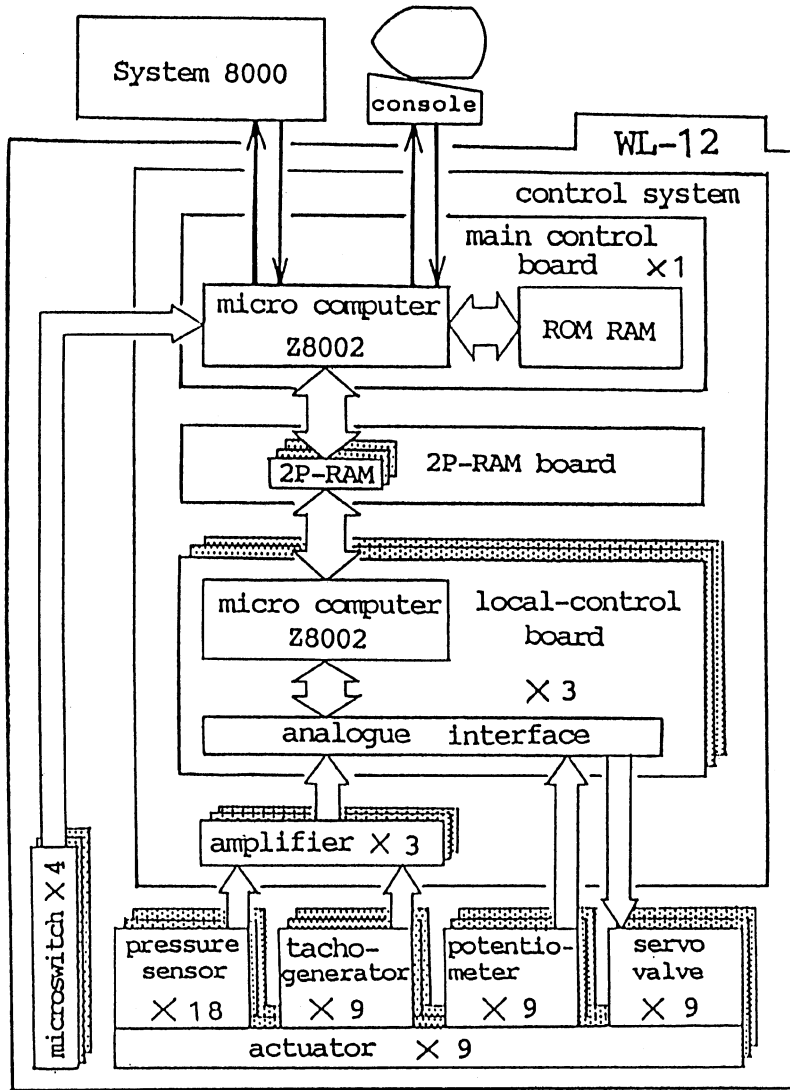


Fig.11 block diagram of control system

Z8002[6] and three local control boards each using 16 bit CPU **Z8002**. Each local control board controls 3 DOFs. The main control board outputs command signals to the local control boards. Communication between the control boards is done with an 8-bit asynchronous parallel transmission using Twoport-RAMs. The communication speed is about 2 [Mbyte/sec].

4 Walking Experiment on Disturbance-Free Floor

As a result of walking experiments on a disturbance-free flat floor, various gaits in dynamic walking stabilized with trunk motion were realized. The minimum walking time was 1.3[sec] a step and the maximum step length was 0.3[m].

The angle and angular velocity responses in the walking experiments are shown in Fig.12. In Fig.12(a), we see the angle responses almost tracked the preset walking patterns. And in Fig.12(b), we see the angular velocity responses on some joints did not reach more than about 150[deg/sec]. I thought that angular velocities in preset walking patterns would go beyond the limitations of the angular velocities of the actuators, so that, for a faster gait, it would be necessary to make preset walking patterns that consider these limitations.

5 Control Method for Walking under External Force

The control method is basically the same as the method for walking without disturbance mentioned above. It consists of two main parts. One is an algorithm that computes balancing motion of the trunk automatically from motion of the lower-limbs, external forces and moments, and a trajectory of the **ZMP** given arbitrarily before the robot begins walking. The other is a program control for walking using a preset walking pattern performed by motion of the lower-limbs and the trunk.

In this section, the algorithm that computes balancing motion of the trunk is described.

5.1 ZMP with External Forces and Moments

The **ZMP** of walking motion of the system modeled in 2.1 on the cartesian coordinate system can be derived as follows:

In the beginning, each vector is defined as shown in Fig.13. Then, we obtain an equation of motion at the arbitrary point **P**, which is obtained by applying **D'Alembert's Principle**.

$$\sum_i m_i (\mathbf{r}_i - \mathbf{P}) \times (\ddot{\mathbf{r}}_i + \mathbf{G}) + \mathbf{T} - \sum_j \mathbf{M}_j - \sum_k (\mathbf{S}_k - \mathbf{P}) \times \mathbf{F}_k = \mathbf{0} \quad (9)$$

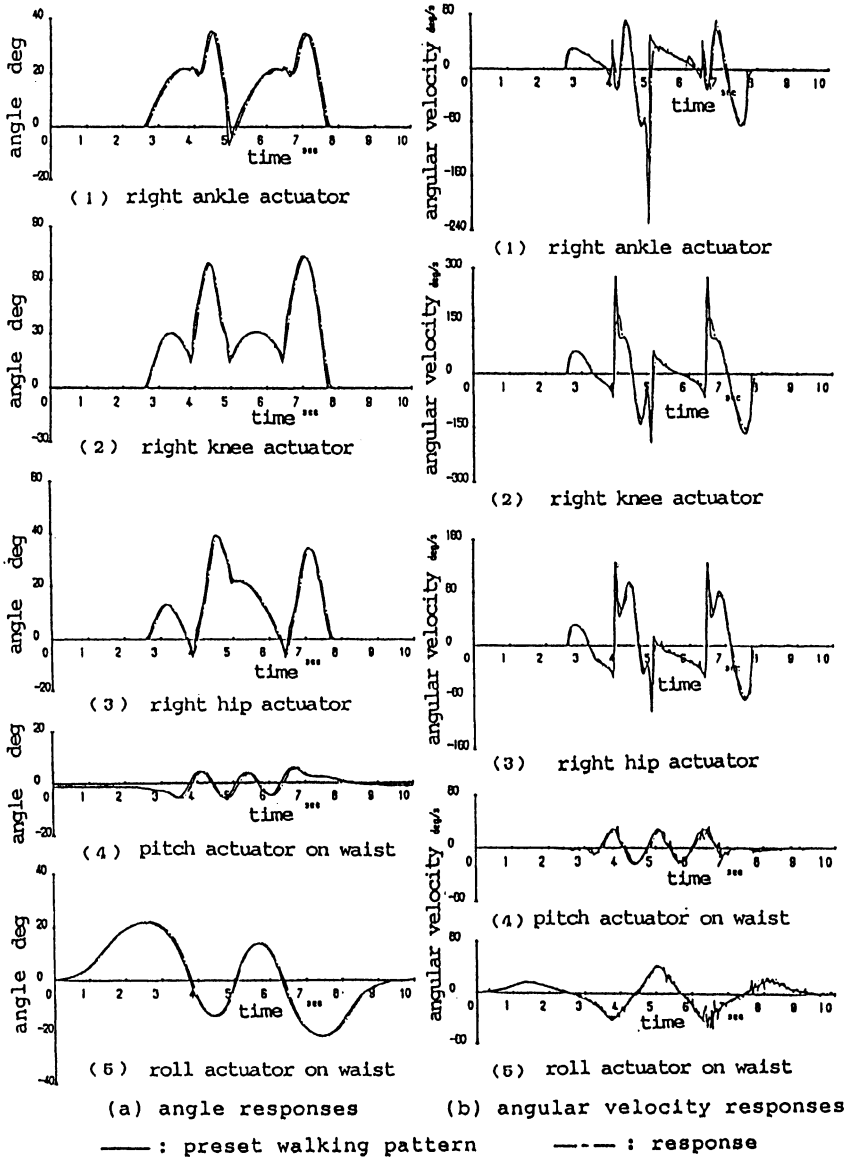
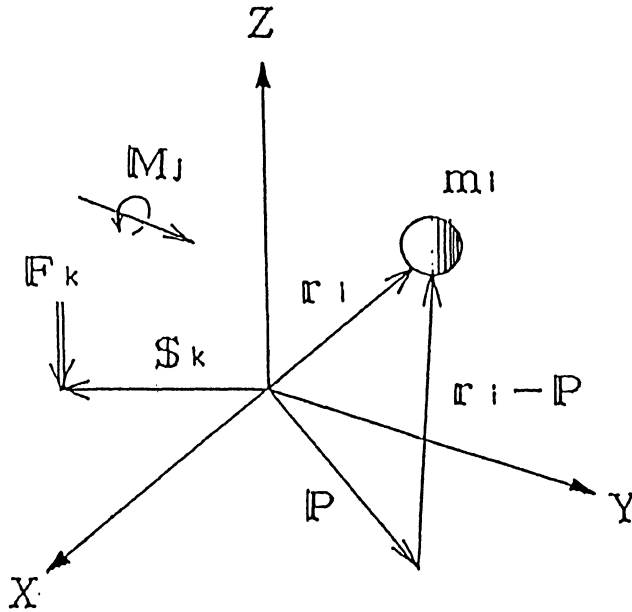


Fig.12 angle and angular velocity responses
 at walking experiment
 (step : 0.3[m] , walking time : 1.3[sec/step])



m_i	:mass of particle i (a scalar)
$r_i = [x_i, y_i, z_i]$	:position vector of particle i
$P = [x_p, y_p, 0]$	:position vector of P
$G = [g_x, g_y, g_z]$	:gravitational acceleration
$T = [T_x, T_y, T_z]$	:total torque acted on P
$M_j = [M_{xj}, M_{yj}, M_{zj}]$	:external moment j
$F_k = [F_{xk}, F_{yk}, F_{zk}]$	:external force k
$S_k = [x_{sk}, y_{sk}, z_{sk}]$	:position vector where external force k is put

Fig.13 definitions of vectors for walking system

By modifying (1), the components of the **ZMP** can be given.

$$\begin{aligned}
 X_{zmp} &= \frac{\sum_{i=0}^n m_i(\ddot{z}_i + g_z)x_i - \sum_{i=0}^n m_i(\ddot{x}_i + g_x)z_i + \sum_j M_{yj} + \sum_k (z_{sk}F_{xk} - x_{sk}F_{zk})}{\sum_{i=0}^n m_i(\ddot{z}_i + g_z) - \sum_k F_{zk}} \\
 Y_{zmp} &= \frac{\sum_{i=0}^n m_i(\ddot{z}_i + g_z)y_i - \sum_{i=0}^n m_i(\ddot{y}_i + g_y)z_i + \sum_j M_{xj} - \sum_k (y_{sk}F_{zk} - z_{sk}F_{yk})}{\sum_{i=0}^n m_i(\ddot{z}_i + g_z) - \sum_k F_{zk}}
 \end{aligned} \tag{10}$$

Let (10) be modified into equations on the moving coordinate. These are divided into two parts as follows. One is the terms of motion of the robot, and the other is the terms of the external forces and moments.

$$\begin{aligned}
 \bar{X}_{zmp} &= \frac{\sum_{i=0}^n m_i(\ddot{\bar{z}}_i + \ddot{\bar{z}}_q + g_z)\bar{x}_i - \sum_{i=0}^n m_i(\ddot{\bar{x}}_i + \ddot{\bar{x}}_q + g_x)(\bar{z}_i + z_q)}{\sum_{i=0}^n m_i(\ddot{\bar{z}}_i + \ddot{\bar{z}}_q + g_z) - \sum_k F_{zk}} : \text{robot motion} \\
 &+ \frac{\sum_j M_{yj} + \sum_k (\bar{z}_{sk}F_{xk} - \bar{x}_{sk}F_{zk})}{\sum_{i=0}^n m_i(\ddot{\bar{z}}_i + \ddot{\bar{z}}_q + g_z) - \sum_k F_{zk}} : \text{external forces and moments} \\
 \bar{Y}_{zmp} &= \frac{\sum_{i=0}^n m_i(\ddot{\bar{z}}_i + \ddot{\bar{z}}_q + g_z)\bar{y}_i - \sum_{i=0}^n m_i(\ddot{\bar{y}}_i + \ddot{\bar{y}}_q + g_y)(\bar{z}_i + z_q)}{\sum_{i=0}^n m_i(\ddot{\bar{z}}_i + \ddot{\bar{z}}_q + g_z) - \sum_k F_{zk}} : \text{robot motion} \\
 &- \frac{\sum_j M_{xj} + \sum_k (\bar{y}_{sk}F_{zk} - \bar{z}_{sk}F_{yk})}{\sum_{i=0}^n m_i(\ddot{\bar{z}}_i + \ddot{\bar{z}}_q + g_z) - \sum_k F_{zk}} : \text{external forces and moments}
 \end{aligned} \tag{11}$$

5.2 Solution for Trunk Motion

We modify (11) into equations in which the terms of the trunk's particle are on the left-hand side and the rest, namely $\alpha(t)$ and $\beta(t)$, are on the right-hand side.

$$\begin{aligned}
 \bar{z}_0(\ddot{\bar{x}}_q + g_x) + \ddot{\bar{z}}_0\bar{x}_{zmp} + (\bar{z}_0 + z_q)\ddot{\bar{x}}_0 - (\ddot{\bar{z}}_0 + \ddot{\bar{z}}_q + g_z)\bar{x}_0 &= \alpha(t) \\
 \bar{z}_0(\ddot{\bar{y}}_q + g_y) + \ddot{\bar{z}}_0\bar{y}_{zmp} + (\bar{z}_0 + z_q)\ddot{\bar{y}}_0 - (\ddot{\bar{z}}_0 + \ddot{\bar{z}}_q + g_z)\bar{y}_0 &= \beta(t)
 \end{aligned} \tag{12}$$

We devide $\alpha(t)$ and $\beta(t)$ into parts as follows. One is the terms of motion of the lower-limbs and the trajectory of the ZMP, and the other is the terms of the external forces.

$$\alpha(t) = \alpha_1(t) + \alpha_2(t) \quad (13)$$

$$\beta(t) = \beta_1(t) + \beta_2(t) \quad (14)$$

where

$$\begin{aligned} \alpha_1(t) = & \left[-\sum_i m_i (\ddot{\bar{z}}_i + \ddot{z}_q + g_z) \bar{x}_{zmp} + \sum_i m_i (\ddot{\bar{z}}_i + \ddot{z}_q + g_z) \bar{x}_i \right. \\ & \left. + \sum_i m_i (\ddot{\bar{x}}_i + \ddot{x}_q + g_x) (\bar{z}_i + z_q) - m_0 (\ddot{x}_q + g_x) \bar{z}_q - m_0 (\ddot{z}_q + g_z) \bar{x}_{zmp} \right] / m_0 \end{aligned}$$

$$\alpha_2(t) = \left[\sum_k F_{zk} \bar{x}_{zmp} + \sum_j M_{yj} + \sum_k \left\{ (z_{sk} + z_q) F_{xk} - x_{sk} F_{zk} \right\} \right] / m_0$$

$$\begin{aligned} \beta_1(t) = & \left[-\sum_i m_i (\ddot{\bar{z}}_i + \ddot{z}_q + g_z) \bar{y}_{zmp} + \sum_i m_i (\ddot{\bar{z}}_i + \ddot{z}_q + g_z) \bar{y}_i \right. \\ & \left. + \sum_i m_i (\ddot{\bar{y}}_i + \ddot{y}_q + g_y) (\bar{z}_i + z_q) - m_0 (\ddot{y}_q + g_y) \bar{z}_q - m_0 (\ddot{z}_q + g_z) \bar{y}_{zmp} \right] / m_0 \end{aligned}$$

$$\beta_2(t) = \left[\sum_k F_{zk} \bar{y}_{zmp} + \sum_j M_{xj} + \sum_k \left\{ (z_{sk} + z_q) F_{yk} - y_{sk} F_{zk} \right\} \right] / m_0$$

I described in 2.2 that $\alpha(t)$ must be known terms. $\alpha_1(t)$ is known because we give motion of the lower-limbs and the trajectory of the ZMP before the robot begins walking. Therefore, $\alpha_2(t)$ must be the known terms. The same thing can be said about $\beta(t)$. When the external forces and moments are known, we can derive the time trajectory of the trunk.

Both equations of (12) interfere with each other and are non-linear differential equations. I already discussed in 2.2 how to obtain a strict periodic solution for the time trajectory of the trunk (see Fig.3).

6 Walking Simulations under External Force

According to the method in chapter 5, as long as the denominator of (10) is positive, external forces and moments can have any direction and strength. However, in our experiments, we adopted to pull the robot backward with a wire. Therefore, adjusting to this, simulations in the case when a rectangular external

force is put on the waist of the robot opposite to the direction it is walking are shown in Fig.13. These figures are the cases when walking the speed or the force strength changes.

In Fig.13, we can see that against the backward external force, the robot compensates moment for the changed **ZMP** by moving its trunk forward at an angle.

The faster the walking speed is, the smaller the trunk's stabilizing motion becomes as the inertial force affects it.

Stabilizing motion against the external force begins a little before the force is applied, and ends a little later when the force returns to zero. I described the reason for this in 2.3.

In this method, as stabilization against an external force is created only by the trunk, the stronger the external force becomes, the larger trunk motion around the pitch axis becomes. As the **WL-12R** has a limited movable angle, the power of the external force which the **WL-12R** can stabilize is limited. Especially when the walking speed is slow, effect of stabilization with inertial force is reduced. Thus its influence is considerable. For example, when the time spent walking is 2.6[sec] a step and the external force is 10[kgf], we can see in the simulation that there are phases where motion of the trunk's pitch axis surpasses the movable angle of its actuator.

7 WL-12R and External Force Generating System

7.1 Biped Walking Robot WL-12R

We improved the **WL-12** mainly in two points and renamed it **WL-12R**. A drawing of the assembly of the **WL-12R** is shown in Fig.14.

The translational joint on the trunk was removed as there had been no need to move the trunk's balancing mass linearly for the effect of the repeating algorithm to solve nonlinear problem. So that, totally the **WL-12R** has 8 DOFs including 2 DOFs on the trunk.

Some parts concerning the robot's structural frame are reinforced using **CFRP** plates to reduce the elastic vibration in walking.

7.2 External Force Generating System

The scene of a walking experiment system under external force is shown in Fig.15. In this system, as described in chapter 6, the direction of the force is limited only in the direction opposite to the walking orientation. And we adopted the system of pulling the robot backward by a wire. We used a **DD** motor, which can be controlled easily and has no clattering with reference to the reduction mechanisms such as the gearing, to generate external forces. We connected a pulley to the motor directly and rolled up the wire to translate the torque of the motor into the tension of the wire.

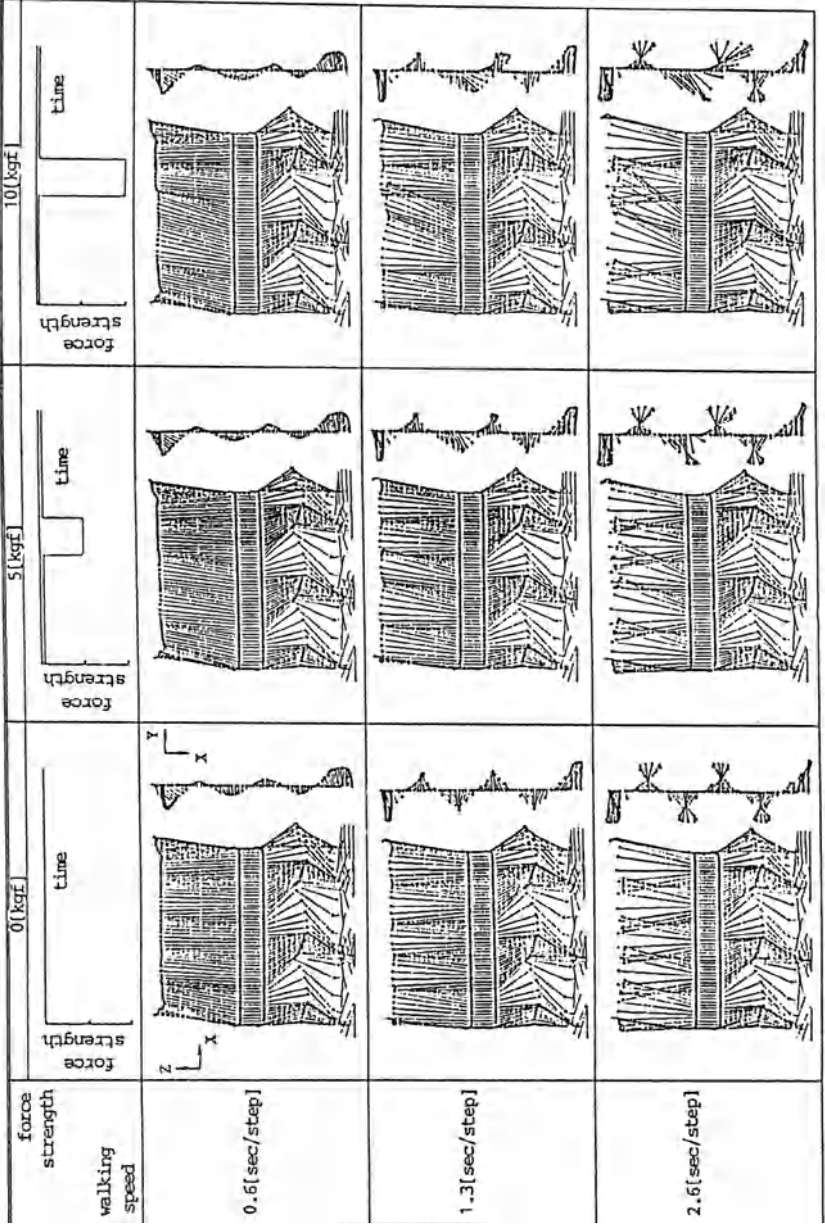


Fig.14 walking simulations

*offsets of the force strength are given as initial tension of wire

As for the sensors, the DD motor is equipped with a pulse-encoder to detect the distance the robot will walk, and a tachogenerator to measure the walking speed. We installed a force sensor to the wire to measure the force which acted on the robot.

8 Walking Experiments under External Force

My coworkers and I performed walking experiments on a flat floor using the WL-12R and the external force generating system. As a result, various gaits in dynamic walking under known external force stabilized with trunk motion were realized. The forces were rectangular and trapezoidal. The maximum force strength was 10[kgf].

The force strength, the distance walked and the walking velocity response produced by the sensors of the external force generating system are shown in Fig.16.

In Fig.16(a), we see the force response almost tracked the preset pattern except for short vibrations. These vibrations were due to the vibration of the wire. In Fig.16(b), we see that the walking velocity response differs considerably from the preset pattern. Excepting the short cycle of vibrations, however, it almost exactly tracks the preset pattern. The reason for this is the same as in the case of the force response.

In the same figure, we see a long cycle of vibrations when the robot stops walking. I think that this is because the robot rocked for a while after it stopped walking, and the sensor recorded its motion.

9 Conclusion

Various gaits in dynamic complete walking on a disturbance-free flat floor stabilized with trunk motion were achieved by the biped walking robot WL-12. The minimum walking period was 0.8[sec] a step and the maximum step length was 0.3[m] step.

Also, various gaits in dynamic complete walking under external force stabilized with trunk motion were achieved by the biped walking robot WL-12R. The maximum force strength was 10[kgf].

As a result, my coworkers and I confirmed the performance of the biped walking robot WL-12(R), and the control methods for robot walking without disturbance and under known external forces and moments on a flat floor were experimentally supported.

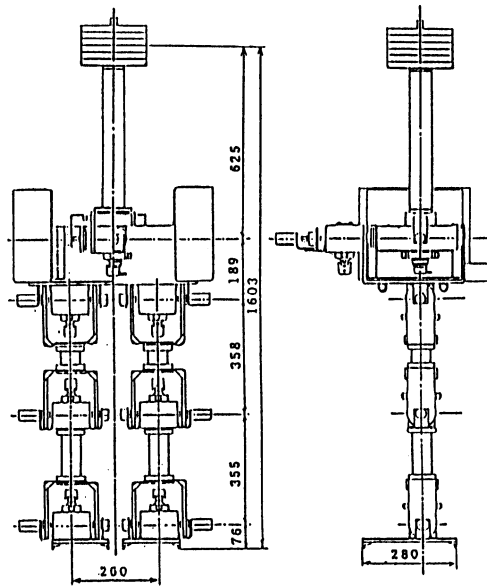


Fig.15 assembly drawing of WL-12

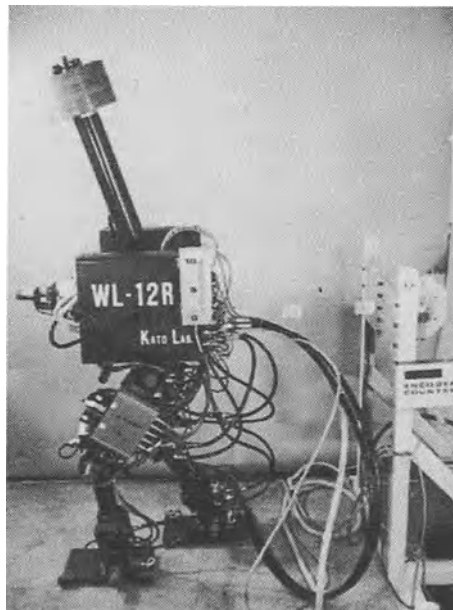
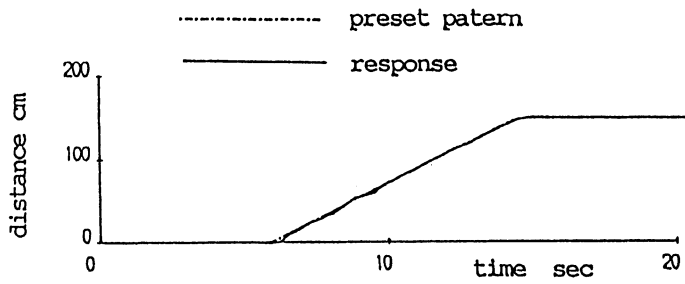
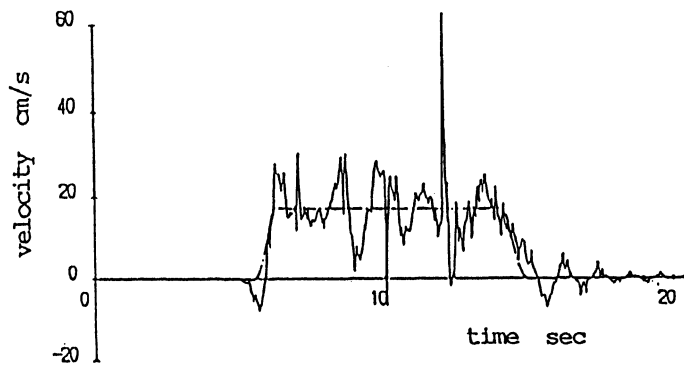


Fig.16 photo of walking experiment



(c) distance walked response



(b) walking velocity response



(a) force strength response

Fig.17 responses at walking experiment
 (step : 0.3[m] , walking time : 1.7[sec/step])

Acknowledgements

The author thanks **KURODA PRECISION INDUSTRIES LTD., MOOG JAPAN LTD. and TORAY INDUSTRIES INC.** for supporting us in developing the **WL-12**.

This study was supported in part by **CASIO SCIENCE PROMOTION FOUNDATION, Japan.**

References

1. Takanishi,A., Ishida,M., Yamazaki,Y. and Kato,I.: The Realization of Dynamic Walking by The Biped Walking Robot WL-10RD, ICAR'85, 1985
2. M.Vukobratovic, A.Frank: On the Stability of Biped Locomotion, IEEE Trans. on Biomedical Engineering, Vol. BME-17, January 1970
3. Takanishi,A., et. al.: Realization of Dynamic Biped Walking Stabilized with TrunkMotion, ROMANSY, 1988
4. Herbert Goldstein: Clasical Mechanics, Addison-Wesley, 1980
5. J.W.Cooley, P.A.W.Lewis, P.D.Welch: The Finite Fourier Transform, IEEE Trans, Audio and Electroacoustics, vol.AU-17, No.2, pp.77-85, June 1969
6. Z8000 Data Manual, Zilog Corporation, 1980

Part 4

Intelligent Motor Control

A New Concept of the Role of Proprioceptive and Recurrent Inhibitory Feedback in Motor Control

Uwe Windhorst

The University of Calgary, Faculty of Medicine, Departments of Clinical Neurosciences and Medical Physiology, 3330 Hospital Drive N.W., Calgary, Alberta, Canada T2N 4N1

Abstract

The mammalian motor control system is a multi-variable, nonlinear and time-varying system. With motor capabilities increasing throughout evolution, the nervous system likely has developed particular solutions to cope with the complex properties of its own executive instruments, in particular muscles. It is proposed here that spinal recurrent inhibition via Renshaw cells and proprioceptive feedback via muscle spindles monitor some important variables of muscle function and adapt spinal neural systems to them.

In mammals, many skeleto-motoneurone pools are a common source of both recurrent inhibition via Renshaw cells and proprioceptive feedback via muscle fibres and proprioceptors. On the other hand, the two feedback pathways exert common actions on a number of spinal neurones including skeleto- and fusi-motoneurones, reciprocal Ia inhibitory interneurones and VSCt cells. This implies that these target neurones receive a *compound information*.

This information is about the basic determinants of skeletal muscle force production. This is determined primarily by the neural input to the muscle fibres, as well as by their length and/or their velocity of length change. The first variable is monitored by Renshaw cells. Muscle fibre length is measured by spindle group Ia and II afferents, dynamic length change by Ia fibres in particular.

An important question is to what extent recurrent inhibition reflects, or predicts, neural muscle force production. This is considered with respect to the two dimensions of motoneurone output: recruitment and rate coding, and their relation to Renshaw cell and force output. Excitatory input to and, to a lesser extent, output from, Renshaw cells show a similar dependency on orderly motoneurone recruitment as does cumulative muscle force output. The static dependencies of Renshaw cell rate and muscle force on motoneurone activation rate also exhibit similar relationships, which for Renshaw cells may be adjustable by proprioceptive feedback of muscle length.

1 Introduction

In some sense, the task of robotics engineers is simpler than that of neurophysiologists. Whilst the former first define their goals and then search for adequate means to achieve them, the latter are confronted with an existing, functioning device, whose parts must be analyzed and functionally interpreted using notions and concepts whose appropriateness is not clear *a priori*.

1.1 General Remarks and Assumptions

1) The control of posture and movement in higher vertebrates is not a simple, but a complex task. Simple models, such as the defunct follow-up servo model, are therefore inadequate to deal with it.

2) The mammalian motor control system is a *multi-variable, nonlinear and time-varying system*. A model has to take account of and represent these properties.

3) The motor control problems to be solved in mammals are not only posed by the constraints originating from external and internal forces and from mechanical interactions between limb segments, but also by those implicated by the very means (e.g. muscles, tendons, joints, etc.) which were evolved by the organism to solve them. *With increasing motor capabilities, the nervous system has developed particular solutions to cope with the properties of its own instruments.*

4) *Proprioceptive feedback* has evolved to monitor the organism's self-generated actions (Sherrington 1906; Evarts 1981). But undoubtedly, it plays various roles in motor control at different levels of the central nervous system. Essentially the same applies to *recurrent inhibition*, this basic circuit being found ubiquitously throughout the neuraxis. However, at spinal level, proprioceptive and recurrent inhibitory feedback exert a common function, i.e., they are involved in the control of muscle force production, particularly in muscles acting across proximal limb joints.

This paper presents a *proposal confined to a subset of roles at the spinal level.*

5) Commonly *descending motor commands* have been envisaged as modulating (even gating) the responses of spinal neural networks to peripheral inputs. In contrast, it is here *postulated that peripheral inputs (including proprioceptive input) serve to modulate such responses to descending commands.* Whereas this does not make much difference physically, it does conceptually (see also Gottlieb and Agarwal 1980).

In the following, a general system description will first be given; then, due to the limited space, the description will focus on some features of muscle force production and recurrent feedback which will illustrate some of the general ideas.

1.2 General System Description

In *mammals* many skeleto- (α, β -) motoneurone pools are a common source of both recurrent inhibition via Renshaw cells and proprioceptive feedback via muscle fibres and proprioceptors. On the other hand, the two feedback pathways exert common actions on a number of spinal neurones including skeleto- and fusi-motoneurones, reciprocal *Ia* inhibitory interneurones and *VSCT* cells (see *Fig. 1*; reviewed in Windhorst 1988). This implies that these target neurones receive a *compound (vector) signal. This carries information about the basic determinants of skeletal muscle state* or, more precisely, its force-producing capacity. Force production is determined primarily by the *neural input to the muscle fibres, their length and/or velocity of length change* (see *Fig. 2*). The first variable is monitored by Renshaw cells, and muscle fibre length is measured by spindle group *Ia* and *II* afferents, dynamic length change by *Ia* fibres in particular (*Fig. 2*).

The *compound information generated by both feedback subsystems* is used to *adapt the parameters of spinal neural networks* (thresholds, parameters of static and dynamic input-output relations and their distributions) to the prevailing

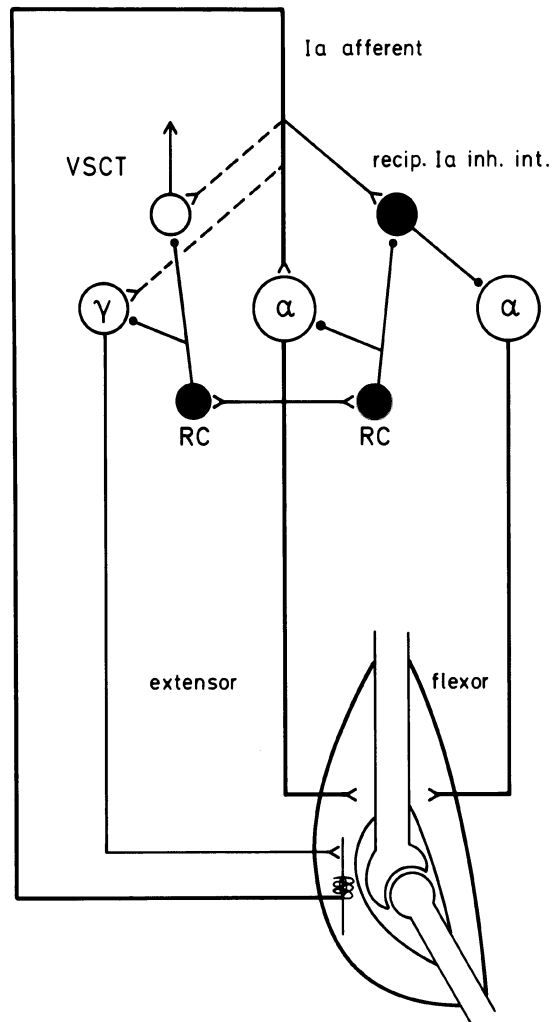


Fig. 1: Schematic comparison of some central neuronal connections involving group Ia afferents (from primary muscle spindle endings) and Renshaw cells. For simplicity, connections from other proprioceptive afferents are not depicted here. Renshaw cells receiving their main excitatory input from extensor α -motoneurons (α) inhibit synergistic α -motoneurons (but not antagonistic α -motoneurons acting around the same joint) and those reciprocal Ia inhibitory interneurons (recip. Ia inh. int.) which receive monosynaptic excitation from extensor Ia fibres and inhibit antagonistic flexor α -motoneurons (right side), and vice versa (not depicted). The same Renshaw cells inhibit extensor γ -motoneurons (left side) and VSCT cells that receive monosynaptic Ia excitation whose exact origin is not yet known, however, and is here assumed to come from extensors (dashed connection).

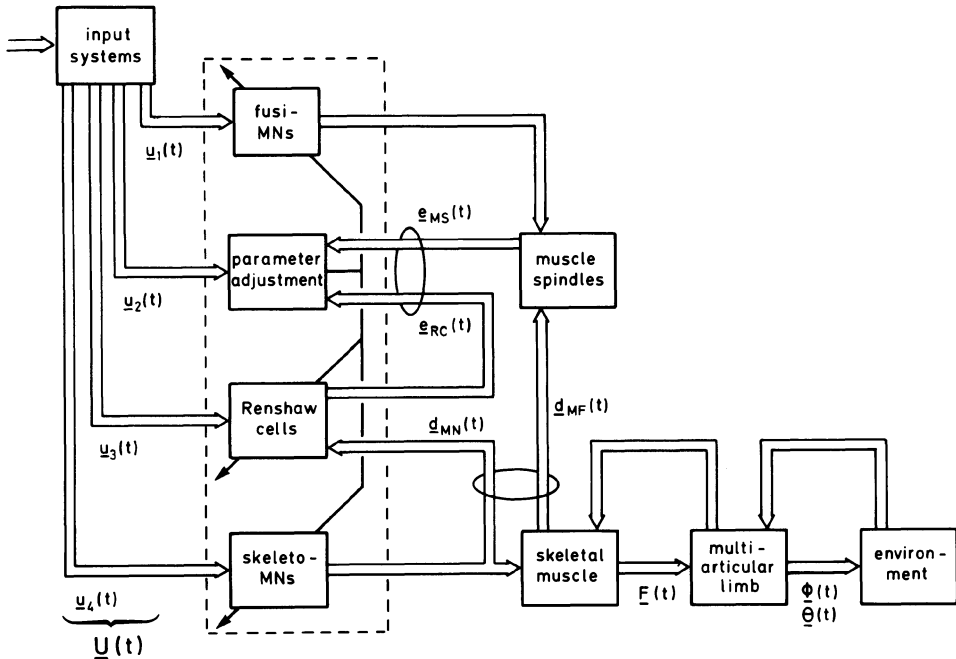


Fig. 2: General scheme of an adaptive motor control system incorporating recurrent inhibition. All pathways are represented by double lines in order to indicate signal vectors. Skeleto-motoneurone (MN) output or muscle input is designated $d_{MN}(t)$ and sensed by Renshaw cells. The muscle output of interest here is designated $d_{MF}(t)$ and is sensed by muscle spindles. Renshaw cells and muscle spindles receive two components of the command (and other) input, $u_3(t)$ and $u_1(t)$, respectively. Their outputs, $e_{RC}(t)$ and $e_{MS}(t)$, are fed back to spinal networks, including skeleto- and fusi-motoneurones, Renshaw cells (and reciprocal Ia inhibitory interneurons, not represented for simplicity), for parameter adjustment, which is symbolized by the oblique arrows originating from the parameter-adjustment subsystem and drawn across the other subsystems. Anatomically, the parameter-adjustment subsystem is not sharply delineated from the other subsystems, as symbolized by the dashed box, but may contain further interneurons. It may thus also receive a separate input signal component $u_2(t)$, different from those to Renshaw cells, fusi-motoneurones and skeleto-motoneurones. The entire input vectors are designated $\underline{U}(t)$. (With permission from Windhorst 1990, his Fig. 2)

muscle mechanical state for *optimization of motor actions* (Fig. 2). Depending on the motor task, this optimization may concern the precision, velocity or smoothness of movement, the stability of posture and movement, or the minimization of energy expenditure and mitigation of the consequences of muscle fatigue.

In general, the force-producing capacity of muscle has to be adapted to a diversity of *motor tasks and their contexts*. These are determined by *descending motor commands* and further *peripheral states* signalled by a variety of segmental afferents (these signals are comprised in the vector $\underline{U}(t)$ in Fig. 2). Commands and afferent information (including spindle group II and Golgi tendon organ signals) thus have an influence, except on skeleto-motoneurones (denoted by $\underline{u}_4(t)$ in Fig. 2), on recurrent inhibition (via pathways circumventing skeleto-motoneurones, denoted by $\underline{u}_3(t)$ in Fig. 2) and proprioceptive feedback (via fusimotor spindle innervation, denoted by $\underline{u}_1(t)$ in Fig. 2). One of the results of this influence is to adjust the *gains* within the state feedback pathways according to the different tasks and contexts. Concerning the feedback from Ia afferents it is therefore clear that, apart from their background excitation, their static and dynamic sensitivities (to length and velocity) should be adjustable by *separate static and dynamic fusimotor neurones*. A similar differential adjustment might occur in the recurrent inhibitory pathway, but has as yet hardly been investigated (Windhorst 1988).

This general concept requires that the organization and properties of both feedback subsystems be matched to a certain extent. The notion of matched inputs to skeleto-motoneurones, fusimotor and Renshaw cells is illustrated in Fig. 3, in relation to descending command signals as well as group II muscle spindle afferents (see also below).

2 Specific Outline

Generally, the behaviour of a system can be described in the form of two sets of equations. The first is a set of

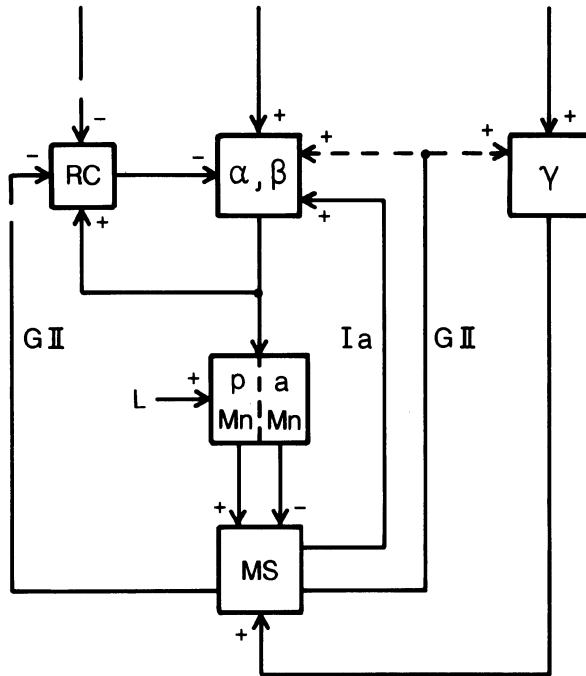


Fig. 3: Simplified block-diagram of the "triple input" to α - and β -motoneurons (middle: α, β), Renshaw cells (left: RC) and γ -motoneurons (right: γ). Descending command inputs are regarded as "matched" if they excite α -, β - and γ -motoneurons, but inhibit Renshaw cells (note that inhibition of these cells disinhibits α - and β -motoneurons), or vice versa. An example is provided by the modulatory influence of muscle length (L) on muscle spindle (MS) afferent signals and recurrent inhibition. The muscle block is divided into two parts. The one labelled "a Mu" (for active muscle) represents the unloading, firing-rate reducing effect of *contraction* on spindles (note sign inversion). The other labelled "p Mu" (for passive muscle) represents the effect of externally generated length changes on spindle discharge, length increase causing firing rate acceleration (note positive sign). These parts are not separable in reality. Spindle afferents have feedback effects on homonymous (and heteronymous) neurones, as follows. As already shown in Fig. 1, Ia fibres (Ia) monosynaptically excite homonymous skeleto-motoneurons, a polysynaptic excitation being represented by the additional interrupted line. Group II (GII) spindle afferents also have a weak monosynaptic connection (dashed line) to these motoneurons. An oligosynaptic inhibition (of extensor motoneurons: interrupted line) occurs when "gated". These afferents have a powerful (oligosynaptic) excitatory action on homonymous γ -motoneurons (e.g., Noth and Thilmann 1980; Appelberg et al. 1983), and a matched inhibitory one on Renshaw cells (Meyer-Lohmann et al. 1976; Fromm et al. 1977; see also Windhorst 1988).

first-order (in general nonlinear) differential equations describing the change in system states as a function of their present values and the system inputs. The second describes the system output. In the following, vector notation (indicated by bold-face letters) is used to emphasize multi-input, multi-output aspects.

2.1 Skeletal Muscle

Specific assumptions and definitions:

1) The *primary output* of skeletal muscle is *force* (denoted by $\mathbf{F}(t)$). This is determined by three major variables: neural excitation (and its history), muscle fibre length and its time derivative (shortening or lengthening velocity).

2) The *neural input* (skeleto-motoneurone output) is a *multi-dimensional signal*, the space being spanned by at least three basis vectors: (a) a *spatial or topographic dimension* (since the signals carried on motor axons are distributed to different portions of the muscle), (b) a *recruitment dimension* (variable number of active motor axons), and (c) a *rate dimension* (variable discharge rate on each motor axon). Hence, it is a vector variable denoted by $\mathbf{d}_{MN}(t)$. Also, skeletal muscle is a time-varying system, i.e., its internal state and output $\mathbf{F}(t)$ depend on the previous history of muscle excitation. This is represented by a delay variable δ in $\mathbf{d}_{MN}(t, \delta)$.

3) According to the well-known *length-tension and force-velocity relations*, the *active force production of each muscle fibre* depends on its length d_{MF} , as well as on the lengthening (positive) or shortening (negative) velocity $v = d_{MF}/dt$ (e.g., Talbot and Gessner 1973; Partridge and Benton 1981; McMahon 1984). It is important to emphasize *muscle fibre length* because this often shows a complicated relation to whole muscle length due to a complex internal muscle architecture (including pinnation; see Gans and Bock 1965; Partridge and Benton 1981; Sacks and Roy 1982). Indeed, muscle fibre length often changes nonlinearly with whole muscle length (Meyer-Lohmann et al. 1974; Windhorst et al. 1976), and pinnation

angle varies in different muscle regions (for examples see Windhorst et al. 1989). Hence, *muscle fibre length and its derivative are (spatially) distributed variables*, thus generally being denoted as vector variables $\underline{d}_{MF}(t)$ and $\underline{d}_{MF}(t)$. They may be considered as *boundary conditions*.

Assume that the dynamic behaviour of the *muscle system* is characterized by a number of state variables $x_i(t)$, $i=1\dots n$. It can then be described by a set of n first-order differential equations:

$$dx_i(t)/dt = M_i\{x_1(t), \dots, x_n(t); \underline{d}_{MN}(t, \delta), \underline{d}_{MF}(t), \underline{v}(t)\}.$$

Since the $x_i(t)$ can be combined in a *state vector* $\underline{x}(t)$, the above system of equations can be written compactly as:

$$d\underline{x}(t)/dt = \underline{M}\{\underline{x}(t), \underline{d}_{MN}(t, \delta), \underline{d}_{MF}(t), \underline{v}(t)\}. \quad (1a)$$

The output (force) vector is given as

$$\underline{F}(t) = \underline{N}\{\underline{x}(t), \underline{d}_{MN}(t, \delta), \underline{d}_{MF}(t), \underline{v}(t)\}. \quad (1b)$$

That is, the temporal change in the (internal system) state vector $\underline{x}(t)$, as expressed by the first equation (1a), as well as the system output $\underline{F}(t)$ (Eq. 1b) are determined by the present state value and by the input vector (muscle excitation $\underline{d}_{MN}(t)$, muscle fibre length $\underline{d}_{MF}(t)$ and velocity $\underline{v}(t)$).

The dependence of active force (F) of a single or a group of parallel fibres on length (d_{MF}) and velocity ($v=d_{MF}/dt$) is usually expressed separately in the two well-known length-tension and force-velocity (Hill) relationships (see, e.g., Talbot and Gessner 1973; Partridge and Benton 1981; McMahon 1984). Carlson (1957) has provided a simple compound mathematical description for whole muscle in which the two above dependencies on length and shortening velocity are represented as additive terms. This approach is here extended and generalized. Thus, assume that $F_O(d_{MN})$ denotes the *isometric* force at the *resting* length d_{MFO} and $v=0$, produced by a given level of neural excitation d_{MN} , that Δd_{MF} is the difference between the actual fibre length d_{MF} and its resting length d_{MFO} . Then:

$$F(d_{MF}, v) = F_O(d_{MN}) - A.(\Delta d_{MF})^2 + B.v/(C+v), \quad (2a)$$

for $v>0$ (lengthening), and

$$F(d_{MF}, v) = F_O(d_{MN}) - A.(\Delta d_{MF})^2 + D.v/(E-v), \quad (2b)$$

for $v < 0$ (shortening). A , B , C , D and E are constants. Note that the second right-hand term which represents the length-dependence is a simple parabola (see also Woittiez et al. 1983) and that the rightmost term in Eq. 2b represents Hill's hyperbola. This formulation separates the three major determinants of muscle force. It is of interest that also the fatiguability of a muscle depends on its length (Fitch and McComas 1985).

2.2 Muscle Spindles

Specific assumptions and definitions:

1) Muscle fibre length $\underline{d}_{MF}(t)$ and velocity $\underline{v}(t)$ (vectorial quantities) are boundary conditions co-determining the internal muscle state for force production.

2) These variables are measured by muscle spindles. With $\underline{y}(t)$ denoting the spindle state, spindle behaviour can be described by the following two state equations:

$$d\underline{y}(t)/dt = \underline{Q}(\underline{y}(t), \underline{d}_{MF}(t), \underline{v}(t), \underline{u}_1(t, \delta)), \quad (3a)$$

$$\underline{e}_{MS}(t) = \underline{P}(\underline{y}(t), \underline{d}_{MF}(t), \underline{v}(t), \underline{u}_1(t, \delta)), \quad (3b)$$

where spindle performance depends on the history of $\underline{u}_1(t, \delta)$ (see, e.g., Brown et al. 1969; Gregory et al. 1987; review in Hulliger 1984).

The response of spindle afferents to large-amplitude length changes is complicated and nonlinear (Hasan 1983) and depends on the pattern of fusimotor input (Boyd 1981; Matthews 1981; Hulliger 1984).

2.3 Renshaw Cells

Specific assumptions and definitions:

1) Renshaw cells monitor the neural motor input to skeletal muscle, i.e., $\underline{d}_{MN}(t)$, rather faithfully (Cleveland 1980), thus providing information about this determinant of force production.

A pair of equations describes the Renshaw cell system as follows:

$$d\mathbf{z}(t)/dt = \mathbf{Q}\{\mathbf{z}(t), \mathbf{d}_{MN}(t), \mathbf{u}_3(t)\}. \quad (4a)$$

$$\mathbf{e}_{RC}(t) = \mathbf{R}\{\mathbf{z}(t), \mathbf{d}_{MN}(t), \mathbf{u}_3(t)\}. \quad (4b)$$

As the neural input to skeletal muscle, the input to Renshaw cells from motor axon collaterals, $\mathbf{d}_{MN}(t)$, varies in three dimensions: a *spatial coordinate, recruitment and rate*.

2.4 Combined Feedback

The recurrent and proprioceptive signals, $\mathbf{e}_{RC}(t)$ and $\mathbf{e}_{MS}(t)$, combine in commonly contacted central neurones to a compound signal, $\mathbf{s}_p(t)$:

$$\mathbf{s}_p(t) = \mathbf{S}\{\mathbf{e}_{RC}(t), \mathbf{e}_{MS}(t)\}, \quad (5)$$

which interacts with other inputs. For instance, in α -motoneurones the signal $\mathbf{d}_{MN}(t)$ is a function of $\mathbf{s}_p(t)$ and $\mathbf{u}_4(t)$:

$$\mathbf{d}_{MN}(t) = \mathbf{T}\{\mathbf{s}_p(t), \mathbf{u}_4(t)\}, \quad (6)$$

where the transformation of $\mathbf{u}_4(t)$ into $\mathbf{d}_{MN}(t)$ is thought to be adjustable by $\mathbf{s}_p(t)$ in a nonlinear way (see Windhorst 1990).

3 Are Renshaw Cells Central Models of Neurally Generated Muscle Force Production?

In the remainder of this paper, it is discussed to which extent Renshaw cell discharge reflects the neurally generated muscle force output. As yet, there is sparse indirect evidence suggesting a similarity of both outputs on recruitment and rate of motoneurone output.

3.1 Recruitment

Muscle. In many steady or slow muscle contractions of increasing force, *motor unit recruitment occurs in an ordered fashion*. Recruitment starts with *S*-type (slowly contracting, fatigue-resistant) motor units, proceeds with *FR*- (fast, fatigue-resistant) and *F(int)*-type (fast, intermediate fatigue resistance) motor units and ends with *FF*-type (fast, fatiguable) motor units (e.g., Burke 1980). Twitch and tetanic contraction forces of the motor units increase in about the

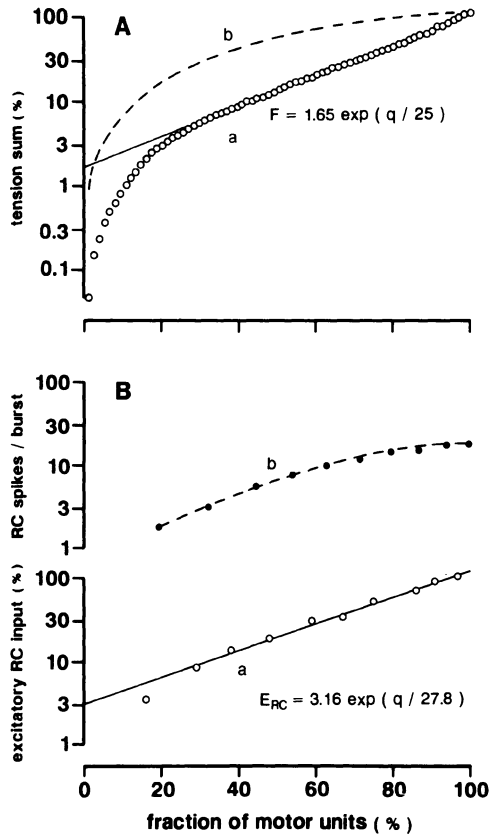


Fig. 4 A-B: Cumulative effects of recruiting increasing percentages of the skeletomotor axons supplying a muscle (abscissa) on muscle force (A) and Renshaw cells (B). Note the logarithmic ordinates. The open circles in part A plot the cumulative muscle tension resulting from summing up the tetanic tensions of the individual motor units in the cat flexor digitorum longus in an order from low to high tetanic tension (*orderly recruitment*). A straight line was fitted (by eye) to the upper part of the data points, indicating an exponential relationship as given (q represents the portion of the motor pool recruited). The dashed line labelled *b* represents a random recruitment model (data points and dashed line replotted from Bagust et al. 1973, their Fig. 1 D). Part B shows, on the same abscissa as in part A, the (normalized) cumulative *excitatory input* to Renshaw cells (E_{RC}) during orderly motor axon recruitment. If Renshaw cells behaved linearly, their *output* (*spike rate*) would obey the same relationship. However, usually they exhibit saturation to increasing excitation from parallel inputs, which has been taken into account in the upper relationship labelled *b* (data extracted from Hultborn et al. 1988, their Figs. 3 and 6a). Note the similarity between the fitted straight lines in parts A and B.

same order. Based on the measurement of isometric tetanic forces of motor units of the cat flexor digitorum longus muscle and assuming a recruitment strictly ordered according to increasing tetanic force, the cumulative muscle force output is depicted by the open circles in *Fig. 4 A*. The upper portion of this relationship can be approximated by a straight line as indicated. The dashed line above (labelled *b*) represents a model of random recruitment (see Bagust et al. 1973). Since actual recruitment does not take place *precisely* according to tetanic motor-unit force (and since some nonlinear interaction may occur between motor units; see Demiéville and Partridge 1980; Niemann et al. 1986), the true cumulative force will probably lie somewhere between the two relationships, but closer to the lower one.

Renshaw cells. The *excitatory input to Renshaw cells* from motor axon collaterals is also differentiated according to motor unit type, increasing from *S-* to *FF*-motoneurons (Cullheim and Kellerth 1978b; Hultborn et al. 1988). Recent results of Hultborn et al. (1988) suggest that this input on average increases with the number of activated motor axons as shown by the open circles (labelled *a*) in *Fig. 4 B* (although there is much variance in the data from individual cells). If Renshaw cells converted this input proportionally into firing rate, the input-output relationship would have the same shape. However, since inputs may interact nonlinearly, the Renshaw cell output follows the relationship represented by the dashed line (labelled *b*). As the nonlinear interaction appears to have been particularly strong due to the testing procedure used by Hultborn et al. (1988), the actual dependence of cumulative Renshaw cell output on the number of recruited motor axons probably lies between the two relationships, possibly closer to the lower one. Note the similarity of the curves labelled *a* in parts *A* and *B*.

3.2 Rate

Muscle. As is well known, the *rate dependence* of muscle force production is nonlinear. *Figure 5* shows several

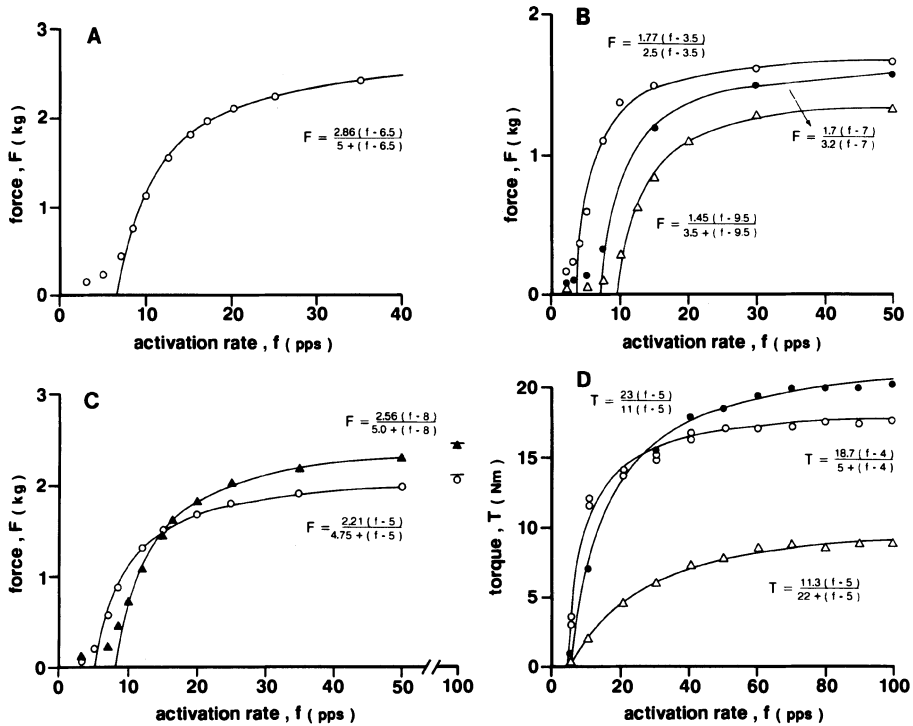


Fig. 5 A-D: Dependence of muscle force and torque on the motor axon activation rate. *Part A:* The open circles plot mean force of the isometric cat soleus muscle as a function of activation rate (data points replotted from Rack and Westbury 1969, their Fig. 7); they are fitted by a rectangular hyperbola disregarding low-rate values as indicated. *Part B:* Effect of ankle joint angle (muscle length) on force-rate relationships of the cat soleus [open circles: joint angle 45° (long muscle); filled dots: joint angle 90°; open triangles: joint angle 120° (short muscle); data replotted from Rack and Westbury 1969, their Fig. 9a]. *Part C:* Force-rate relationships of the cat soleus muscle during isometric (open circles) and dynamic lengthening (filled triangles) at a joint angle of 70° (data replotted from Joyce et al. 1969, their Fig. 4). *Part D:* Effect of ankle joint angle on the relationship between ankle dorsiflexor torque and rate of supramaximal stimulation of the tibial anterior muscle in man [open circles: joint 30° plantarflexed (long muscle); filled dots: joint 5° plantarflexed; open triangles: joint 20° dorsiflexed (short muscle); data replotted from Marsh et al. 1981, their Fig. 5 B]. Note the shortening of k with increasing muscle length in both parts B and D.

examples. The measurements in *Fig. 5 A-B* were made on the *slow* isometric cat soleus muscle. The data show the frequently observed *S-shaped* relationship between activation rate and force. Usually, however, the lowest regular firing rates of motor units at their recruitment are between ca. 5 and 12 *pps* (Freund 1983). Hence, the points at the very low rates in *Fig. 5 A* and *B* are not important physiologically. For this and another (see below) reason, I have fitted the data with rectangular hyperbolas of the form

$$F = \frac{b \cdot (f - f_a)}{j + (f - f_a)}, \quad f > f_a \quad (7)$$

where F denotes muscle force, f the activation rate, f_a the abscissa intersection, b the saturation constant (maximal force output approximated at infinite rate), and j the rate at which half the saturation is attained. *The steepness and location on the f -axis of the steep part of the rate-force relationship is important because it determines the range of most effective rate modulation.*

Figure 5 B suggests that (1) the abscissa intersection (f_a) and (2) the semi-saturation constant j decline with increasing muscle length (joint angle; see legend). The latter constant (j) also declines if, in man, the rate-dependence of the ankle dorsiflexor torque (T) exerted by the *faster* tibial anterior is determined at increasing plantarflexion (see legend). Very probably, this length dependence results from a slowing of muscle twitch (particularly the relaxation phase) with an increase in muscle length (Rack and Westbury 1969; Marsh et al. 1981). That is, j increases with decreasing contraction speed. Also, *Fig. 5 C* for the cat soleus muscle shows that the rate-force relationship also depends on *dynamic* muscle lengthening. Hence, this relationship is strongly co-determined by peripheral factors, such as muscle length and velocity. The question then arises whether the Renshaw cell system can at all reflect this situation.

Renshaw cells. The relationship between the activation rate of motor axons (f) and the discharge rate of Renshaw

cells (r) can also be described by a rectangular hyperbola (Cleveland 1980; Cleveland et al. 1981):

$$r = \frac{c \cdot f}{k + f} \quad (8)$$

where c is the maximal (saturation) Renshaw cell rate attainable with that input, and k denotes the semi-saturation input rate. An example from our own data is illustrated in Fig. 6 A. The distribution of k found by Cleveland (1980) is shown in Fig. 6 B. It appears as if, on average, k tended to be longer than j (cf. Fig. 5), the more so as the physiologically relevant relation between input (f) and output (r) rates starts at $f > 0$ pps (see above). That is, as illustrated in Fig. 6 C, k' becomes longer if determined for the relationship above, say, $f = 6$ pps.

However, it must be taken into account that the static Renshaw cell input-output relationship is not constant, but is subject to modulating influences (see Fig. 3). There is much evidence that command signals for muscle activation, except for being directed to skeleto-motoneurones, also influence Renshaw cells via independent pathways (reviewed in Windhorst 1988). Hultborn and Pierrot-Deseilligny (1979) proposed that Renshaw cells coupled to the human soleus muscle are progressively depressed (from an initially facilitated state) during a continuing ramp-like isometric muscle contraction. Hence, these cells receive augmenting excitatory input (from soleus motor axon collaterals) and inhibitory input (from independent descending pathways; see Fig. 3). Assume that, as a result, Renshaw cell firing is dynamically depressed such that the saturation constant, c , linearly decreases with increasing activation rate, r , during augmenting contraction (recruitment is here disregarded). As an example, for the lower curve in Fig. 6 D, c decreases from a value of $c = 200$ pps at $r = 0$ pps to $c = 150$ at $r = 50$ pps. The constant k will then appear to be shorter than it would be without the depressing effect. Moreover, this effect may be enhanced by the positive linear correlation between c and k across cells (Cleveland 1980). Finally, k might also be depressed directly by the

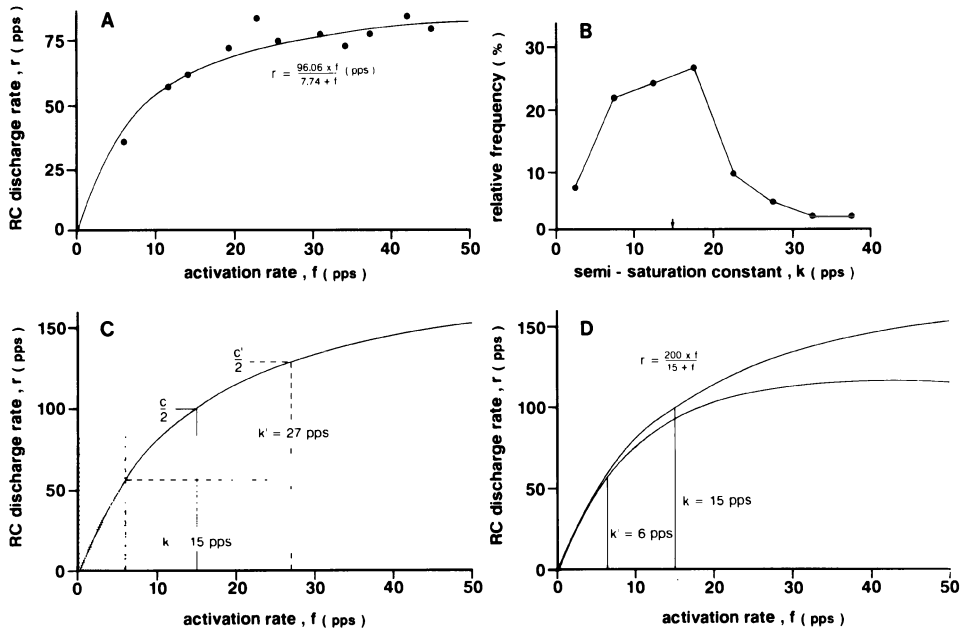


Fig. 6 A-D: Dependence of static Renshaw cell firing rates on motor axon activation rates. Part A displays our own measurements from a cat Renshaw cell activated by supramaximal stimulation of motor axons in the posterior biceps nerve (dorsal roots L_6-S_1 cut) at constant but different rates (solid dots). The dots were approximated by a rectangular hyperbola passing through the origin as indicated. The plot in part B shows the distribution of the semi-saturation constants, k , as found by Cleveland (1980; his Fig. 5, with permission); 41 measurements on 32 cells; the arrow on the abscissa indicates the mean value. Part C illustrates the effect on k of disregarding activation rates lower than $r=6$ pps (stippled area). The characteristic values of the complete original rectangular hyperbola passing through the origin ($c=200$ pps, $k=15$ pps) become larger when determined for the same relationship above $r=6$ pps (dashed lines). Part D shows the effect, on the same rectangular hyperbola as depicted in part C (upper curve in D), of linearly decreasing c with increasing r (from $c=200$ pps at $r=0$ pps to $c=150$ pps at $r=50$ pps). The apparent semi-saturation constant, k , thereby decreases from $k=15$ pps to about $k=6$ pps.

descending command signal. Whether such effects occur is unknown and remains to be investigated.

Also, as illustrated in *Fig. 3*, group *II* afferents from secondary muscle spindle endings possibly inhibit homonymous Renshaw cells (Meyer-Lohmann et al. 1976; Fromm et al. 1977). If this has the same effects on *c* and *k* as described above for the descending command signal, this could adjust *k* to muscle length, in a way mimicking the dependence of *j* on muscle length (*Fig. 5*). The precise quantitative relationships have still to be determined in the same muscle of the same animal.

Two points should be added: (1) At least as important as the adjustment of static input-output relations is that of *dynamic characteristics*; (2) The adjustment of recurrent inhibition also affects the *signal transfer properties of motoneurones*.

4 Concluding Remarks

The preceding exemplifying and hypothetical considerations suggest the intriguing possibility that (a) recurrent inhibition mediated via Renshaw cells is a central model of neurally generated muscle force production, (b) the parameters of this model (e.g., *c* and *k*) may be adjusted, both via descending commands ($\underline{u}_3(t)$ in *Fig. 2*), to special task requirements, or, via proprioceptive feedback (in this case from group *II* spindle afferents, or other afferents, such as those from joints), to peripheral conditions. This is no unique occurrence since the stretch reflex dynamics (Weiss et al. 1986) and the relative neural activation of different muscles or muscle parts (Jongen et al. 1989) depend on joint angle (muscle length), as well.

For lack of space - and of data - the cooperative interaction - in central neurones - of recurrent inhibition, yielding a prediction of neurally generated muscle force, and of spindle afferents, yielding state feedback about two further determinants of muscle force (muscle fibre length and its change), could not been treated here at any length. This remains a task for the future.

4.1 The Degree of Match

One of the central hypotheses of this paper is that recurrent inhibition and proprioceptive feedback provide a compound feedback to spinal networks signalling the basic determinants of muscle force production. A prerequisite appears to be a certain match between structural and functional characteristics of the two systems. An extensive comparison in this regard has been given elsewhere (Windhorst 1988). The extent to which properties of the two systems need to be matched is generally not easy to determine because, firstly, recurrent inhibition and proprioceptive feedback monitor different parameters (*Ia* feedback having to mirror the complicated mechanical periphery) and, secondly, the match would be expected to depend on the specific muscle systems under study. But there are differences that need a few comments.

1) The central connections of Renshaw cells and spindle *Ia* afferents onto motoneurons are similar, albeit not completely isomorphic (Baldissera et al. 1981; Hamm 1990). In the cat hindlimb, recurrent inhibition interconnects more motoneurone pools than do *Ia* afferents, which appears to be a counterargument to the required match. However: (a) Such a match *cannot be expected* to be completely isomorphic (see above). (b) The distribution of recurrent inhibition (from motor axons to motoneurons) cannot be fairly compared with that of *Ia* afferent connections because the latter represent only the second part of the feedback loop, the first being the mechanical coupling between motor units and spindles. This is usually not taken into account in such comparisons. (c) Not all the interconnections may be active all the time (being subject to various modulating influences; see above). (d) Spindle feedback is also provided by group II afferents which have a wider (and more complicated) distribution than do *Ia* afferents (see Windhorst 1988).

2) Recurrent inhibition is absent in some limb muscles, in particular in distal foot and hand muscles (human hand muscles: Person and Kozhina 1978; cat "hand" muscles: Hahne et

al. 1988; cat short plantar muscles: Cullheim and Kellerth 1978a). Recurrent inhibition thus appears to be more prevalent in proximal limb muscles (Baldissera et al. 1981). The functional reason behind this may be that muscles around proximal joints have to move larger *inertia* than do distal muscles which simply move the digits. Force control to cope with inertial problems may thus be a major issue with which recurrent inhibition may be functionally associated.

The connectivity of both spindle group *II* and Golgi tendon organ afferents is very wide and diffuse, including excitatory and inhibitory effects (e.g., McCrea 1986). This justifies to assume that secondary spindle endings (in addition) and Golgi tendon organs play a role in motor control differing from that played by recurrent inhibition and spindle *Ia* afferents, probably in providing muscle length and tension signals for the coordination of activities in many limb muscles. In particular, as exemplified for spindle group *II* afferents in *Fig. 3*, they could in turn adapt parameters in the above two systems. In other words, spindle group *II* and tendon organ afferents form a system distinct from that constituted by recurrent inhibition and *Ia* afferent feedback.

4.2 Common Principles

Several features put the present suggestion into a wider context, possibly pointing to common design principles in the nervous system.

1) Skeletal force production is determined by a complex nonlinear interaction of various factors. It is here proposed that the interaction is *decomposed* into these (or some of these) factors by Renshaw cells and *Ia* afferents (with a possible contribution from group *II* spindle afferents). By convergence of Renshaw cell and *Ia* outputs on several spinal neural systems, the separate signals are *re-composed*, probably in a nonlinear fashion. Similarly, many sensory systems carry feature-specific information on separate channels which then, at some point, converge on *integrative* systems.

2) This re-composition creates complex *representations in multi-dimensional spaces*. Populations of Renshaw cells monitor

the activity of many synergistic motoneurons in a way that apparently is related to their proximity and/or functional similarity and capacity of force production. The vectorial output of many Renshaw cells will then represent the expected vectorial force (or torque) output due to neural drive. Similarly, the vectorial output of spindle Ia afferents represents vectorial measures of muscle fibre length and velocity (which need not be exactly collinear with the recurrent vector). The possibly nonlinear summation of these vectorial quantities in spinal postsynaptic neurons will create a new *mixed and more global representation* (Hasan 1990). Similarly, mixed vectorial quantities are also encoded by motor cortical cells in primates. Cells whose firing is related to shoulder and elbow movements display a broadly tuned discharge response to various directions of hand movement (Georgopoulos et al. 1986, 1989; Kalaska et al. 1989; Kalaska 1990). That is, each neurone has a "preferred direction" with a large variation around it. The direction of hand movement can be fairly well predicted by a vectorial sum of a whole population of motor cortical cells (Georgopoulos et al. 1986, 1989). In addition to direction, many cells in cortical area 4 also code the direction of the load to be moved. Apparently this load-sensitivity is characteristic of neurons related to muscles across the shoulder which control the trajectory of the reaching movement and thus have to cope with arm inertia (Kalaska 1990).

References

1. Appelberg, B., Hulliger, M., Johansson, H., Sojka, P.: Actions on γ -motoneurons elicited by electrical stimulation of group II muscle afferent fibres in the hind limb of the cat. *J. Physiol. (Lond.)* 335: 255-273 (1983)
2. Bagust, J., Knott, S., Lewis, D.M., Luck, J.C., Westerman, R.A.: Isometric contractions of motor units in a fast twitch muscle of the cat. *J. Physiol. (Lond.)* 231: 87-104 (1973)
3. Baldissera, F., Hultborn, H., Illert, M.: Integration in spinal neuronal systems. In: *The nervous system. Motor control* (V.B. Brooks, ed.), pp. 509-595 (*Handbook of physiology, Sect. 1, Vol. II*) Bethesda: Amer. Physiol. Soc. 1981

4. Boyd, I.A.: The action of the three types of intrafusal fibre of isolated cat muscle spindles on the dynamic and length sensitivities of primary and secondary sensory endings. In: Muscle receptors and movement (A. Taylor, A. Prochazka, eds.), pp. 17-32. London:Macmillan 1981
5. Brown, M.C., Goodwin, G.M., Matthews, P.B.C.: Aftereffects of fusimotor stimulation on the response of muscle spindle primary afferent endings. *J. Physiol. (Lond.)* 205: 677-694 (1969)
6. Burke, R.E.: Motor unit types: functional specializations in motor control. *Trends Neurosci.* 3: 255-258 (1980)
7. Carlson, F.D.: Kinematic studies on mechanical properties of muscle. In: Tissue elasticity (J.W. Remington, ed.), pp. 55-72. Washington DC: Amer. Physiol. Soc. 1957
8. Cleveland, S.: Verarbeitung spinal-motorischer Ausgangssignale durch die Renshaw-Zellen. Univ. Düsseldorf: Med. Habil.-Schrift, 1980
9. Cleveland, S., Kuschmierz, A., Ross, H.-G.: Static input-output relations in the spinal recurrent inhibitory pathway. *Biol. Cybern.* 40: 223-231 (1981)
10. Cullheim, S., Kellerth, J.-O.: A morphological study of the axons and recurrent axon collaterals of cat α -motoneurons supplying different hind-limb muscles. *J. Physiol. (Lond.)* 281:285-299 (1978a)
11. Cullheim, S., Kellerth, J.-O.: A morphological study of the axons and recurrent axon collaterals of cat α -motoneurons supplying different types of muscle unit. *J. Physiol. (Lond.)* 281: 301-313 (1978b)
12. Demiéville, H.N., Partridge, L.D.: Probability of peripheral interaction between motor units and implications for motor control. *Amer. J. Physiol.* 238 (Regulatory Integrative Comp. Physiol. 7): R119-R137 (1980)
13. Evarts, E.V.: Sherrington's concept of proprioception. *Trends Neurosci.* 4: 44-46 (1981)
14. Fitch, S., McComas, A.: Influence of human muscle length on fatigue. *J. Physiol. (Lond.)* 362: 205-213 (1985)
15. Freund, H.-J.: Motor unit and muscle activity in voluntary motor control. *Physiol. Rev.* 63: 387-436 (1983)
16. Fromm, C., Haase, J., Wolf, E.: Depression of the recurrent inhibition of extensor motoneurons by the action of group II afferents. *Brain Res.* 120: 459-468 (1977)
17. Georgopoulos, A.P., Schwartz, A.B., Kettner, R.E.: Neuronal population coding of movement direction. *Science* 233: 1416-1419 (1986)
18. Georgopoulos, A.P., Lurito, J.T., Petrides, M., Schwartz, A.B., Massey, J.T.: Mental rotation of the neuronal population vector. *Science* 243: 234-236 (1989)
19. Gans, C., Bock, W.J.: The functional significance of muscle architecture - a theoretical analysis. *Ergeb. Anat.* 38: 115-142 (1965)
20. Gottlieb, G.L., Agarwal, G.L.: Response to sudden torques about the ankle in man. II. Postmyotatic reactions. *J. Neurophysiol.* 43: 86-101 (1980)
21. Gregory, J.E., Morgan, D.L., Proske, U.: Changes in size of the stretch reflex of cat and man attributed to aftereffects in muscle spindles. *J. Neurophysiol.* 58: 628-640 (1987)

22. Hahne, M., Illert, M., Wietelmann, D.: Recurrent inhibition in the cat distal forelimb. *Brain Res.* 456: 188-192 (1988)
23. Hamm, T.M.: Recurrent inhibition to and from motoneurons innervating the flexor digitorum and flexor hallucis longus muscles of the cat. *J. Neurophysiol.* (1990) (in press)
24. Hasan, Z.: A model of spindle afferent response to muscle stretch. *J. Neurophysiol.* 49: 989-1006 (1983)
25. Hasan, Z.: Biomechanics and the study of multijoint movements. In: *Freedom to move: dissolving boundaries in motor control* (Humphrey, D.R., Freund, H.-J., eds). Dahlem Konferenzen. Chichester: John Wiley&Sons (1990) (in press)
26. Hulliger, M.: The mammalian muscle spindle and its central control. *Rev. Physiol. Biochem. Pharmacol.* 101: 1-110 (1984)
27. Hultborn, H., Pierrot-Deseilligny, E.: Changes in recurrent inhibition during voluntary soleus contractions in man studied by an H-reflex technique. *J. Physiol. (Lond.)* 297: 229-251 (1979)
28. Hultborn, H., Lipski, J., Mackel, R., Wigström, H.: Distribution of recurrent inhibition within a motor nucleus. I. Contribution from slow and fast motor units to the excitation of Renshaw cells. *Acta Physiol. Scand.* 134: 347-361 (1988)
29. Jongen, H.A.H., Denier van der Gon, J.J., Gielen, C.C.A.M.: Inhomogeneous activation of motoneurone pools as revealed by co-contraction of antagonistic human arm muscles. *Exp. Brain Res.* 75: 555-562 (1989)
30. Joyce, G.C., Rack, P.M.H., Westbury, D.R.: The mechanical properties of cat soleus muscle during controlled lengthening and shortening movements. *J. Physiol. (Lond.)* 204: 461-474 (1969)
31. Kalaska, J.F.: What parameters of reaching are encoded by discharges of cortical cells? In: *Freedom to move: dissolving boundaries in motor control* (Humphrey, D.R., Freund, H.-J., eds). Chichester: John Wiley&Sons (1990) (in press)
32. Kalaska, J.F., Cohen, D.A.D., Hyde, M.L., Prud'homme, M.: A comparison of movement direction-related activity versus load direction-related activity in primate motor cortex, using a two-dimensional reaching task. *J. Neurosci.* 9: 2080-2102 (1989)
33. Marsh, E., Sale, D., McComas, A.J., Quinlan: Influence of joint position on ankle dorsiflexion in humans. *J. Appl. Physiol.* 51: 160-167 (1981)
34. Matthews, P.B.C.: Muscle spindles: their messages and their fusimotor supply. In: *The nervous system* (V.B. Brooks, ed.), *Handbook of physiology*, Vol. II, Part 1, pp. 189-228. Bethesda: Amer. Physiol. Soc. (1981)
35. McCrea, D.A.: Spinal cord circuitry and motor reflexes. *Exercise Sport Sci. Rev.* 14: 105-141 (1986)
36. McMahon, T.A.: *Muscles, reflexes, and locomotion*. Princeton: Princeton University Press 1984
37. Meyer-Lohmann, J., Henatsch, H.-D., Benecke, R., Hellweg, C.: Muscle stretch and chemical muscle spindle excitation: effects on Renshaw cells and efficiency of recurrent inhibition. In: *Understanding the stretch reflex* (S. Homma, ed.), *Progr. Brain Res.*, Vol. 44, pp. 223-233. Amsterdam-Oxford-New York:Elsevier (1976)

38. Meyer-Lohmann, J., Riebold, W., Robrecht, D.: Mechanical influence of the extrafusal muscle on the static behaviour of deafferented primary muscle spindle endings in cat. *Pflügers Arch.* 352: 267-278 (1974)
39. Niemann, U., Windhorst, U., Meyer-Lohmann, J.: Linear and nonlinear effects in the interactions of motor units and muscle spindle afferents. *Exp. Brain Res.* 63: 639-649 (1986)
40. Noth, J., Thilmann, A.: Autogenetic excitation of extensor-motoneurons by group II muscle afferents in the cat. *Neurosci. Lett.* 17: 23-26 (1980)
41. Partridge, L.D., Benton, L.A.: Muscle, the motor. In: *The nervous system* (V.B Brooks, ed.), *Handbook of physiology*, Vol. II, Part 1, pp. 43-106. Bethesda: Amer. Physiol. Soc. 1981
42. Person, R.S., Kozhinam G.V.: Study of orthodromic and antidromic effects of nerve stimulation on single motoneurons of human hand muscles. *EMG Clin. Neurophysiol.* 18: 437-456 (1978)
43. Rack, P.M.H., Westbury, D.R.: The effects of length and stimulus rate on tension in the isometric cat soleus muscle. *J. Physiol. (Lond.)* 204: 443-460 (1969)
44. Sacks, R.D., Roy, R.R.: Architecture of the hind limb muscles of cats: functional significance. *J. Morphol.* 173: 185-195 (1982)
45. Sherrington, C.S.: On the proprioceptive system, especially in its reflex aspects. *Brain* 29: 467-482 (1906)
46. Talbot, S.A., Gessner, U.: *Systems physiology*. New York-London-Sidney-Toronto: Wiley 1973
47. Weiss, P.L., Kearney, R.E., Hunter, I.W.: Position dependence of stretch reflex dynamics at the human ankle. *Exp. Brain Res.* 63: 49-59 (1986)
48. Windhorst, U.: How brain-like is the spinal cord? Interacting cell assemblies in the nervous system. Berlin-Heidelberg-New York-London-Paris-Tokyo: Springer-Verlag 1988
49. Windhorst, U.: Do Renshaw cells tell spinal neurones how to interpret muscle spindle signals? In: *Afferent control of posture and locomotion* (J.H.J. Allum, M. Hulliger, eds.), *Progress in Brain Research*, Vol. 55. Amsterdam: Elsevier (1990) (in press)
50. Windhorst, U., Hamm, T.M., Stuart, D.G.: On the function of muscle and reflex partitioning. *Beh. Brain Sci.* (1989) (in press)
51. Windhorst, U., Schmidt, J., Meyer-Lohmann, J.: Analysis of the dynamic responses of deafferented primary muscle spindle endings to ramp stretch. *Pflügers Arch.* 366: 233-240 (1976)
52. Woittiez, R.D., Huijing, P.A., Rozendal, R.H.: Influence of muscle architecture on the length-force diagram of mammalian muscle. *Pflügers Arch.* 399: 275-279 (1983)

Analogic Models for Robot Programming

R.Zaccaria, P.Morasso and G.Vercelli

DIST - University of Genoa - Via Opera Pia, 11A - 16145 Genova, Italy

Abstract. Planning robot movements in an unstructured environment is investigated from the point of view of common sense reasoning as the simulation of a complex dynamical system. We first show how motor redundancy can be solved by simulating passive motion (what we call *Passive Motion Paradigm*). We then extend this concept in terms of *Abstract Force Fields*, that allow expression of a variety of problems in addition to trajectory formation (force/position control, obstacle avoidance, concurrent tasks, etc.). Furthermore, we define the notion of *Hybrid Motor Schema* as the composite ensemble of an analogic component (which consists of the "mental" simulation of model dynamics) and a symbolic component (which contains rules and methods for instantiating the analogic component in a given context: goal, environment, expectations, constraints, ..). The paper then describes in more detail two specific analogic models in motion planning. The first is a connectionist formulation of motor planning for a redundant anthropomorphic structure (M - nets). The second is a general framework for planning sequences of actions for assembling objects. Simulation results of preliminary implementations are presented.

1. Introduction

Planning robot movements in an uncertain, unstructured environment is a typical problem of dealing with the real world. This kind of planning cannot be reduced to trajectory formation only; it also needs to deal with redundancy, complex 3D modelling, and knowledge-based representation. These are all aspects of the same problem, in the sense that the complexity of planning should be analyzed in a global manner, and then distributed to a set of concurrent computational models.

The present paper reports a line of research that approaches this problem taking into account concepts of common sense reasoning. Common sense reasoning is inspired by some hypotheses (Steels, 1988):

- a. models are just approximations of the real world;
- b. reality requires multiple-level representations;
- c. complexity involves both symbolic and analogic computations;
- d. problems are often "ill-posed", and the solution space is practically infinite and constraints tend to be implicit;
- e. world knowledge is rarely complete;
- f. non-linear dynamics is a powerful problem solving tool.

Summing up, we can single out two components in a common sense system:

1. An analogic component, which consists of the "mental" simulation of the dynamics of a model which has some degree of analogy with a relevant chunk of the real world;

2. A symbolic component, which contains rules and methods for instantiating the analogic component in a given context (goal, environment, expectations, constraints, ..).

The ensemble of the two components is something which obviously must have some affinity with the notion of schema/frame/script/..., the peculiarity being given by the analogic component: therefore, we wish to call it *hybrid schema*. The rest of the paper is devoted to explain what we mean by this in the specific domain of robot planning and control (P&C), while it also shows some preliminary simulation results.

The general concepts are introduced in sections 2 to 4. In section 5 some examples of robot motions are discussed. In section 6 a specific case of a neural model for a redundant robot is described. In section 7, a framework for using analogic models in planning sequences of actions is presented.

2. Redundancy and the Passive Motion Paradigm

Redundancy is one of the key aspects to view when studying the complexity of motor P&C for both the biological and artificial cases. Redundant systems are characterized by an excess of degrees of freedom with respect to the number of parameters required for specifying a task. For example, only six kinematic parameters are needed to express the location of a rigid tool in the workspace. In contrast, the degrees of freedom of the human arm are at least seven not including the fingers and the shoulder-girdle complex.

Redundancy of motor systems introduces some complex mathematical problems since the kinematic equations do not admit unique well-defined inverse solutions. However, the effort of facing these difficulties is paid back by increased flexibility: motor redundancy endows the motor planner with a rich repertoire of coordinated strategies (instead of a unique solution) for any single kinematic task. In a sense, redundancy appears to be the key to versatility of human movements and for this reason it attracts an increasing interest in robotics research.

Several investigators of biological systems have focussed on the role of muscle mechanical properties in motor control, suggesting that muscles are mechanically analogous to "tunable" springs: i.e., they are characterized by a set of integrable functions between length and tension at steady-state. Neural input to each muscle selects a particular function out of this set. The equilibrium position and the stiffness of a joint is then defined, for any given setting of muscle activations, as the position at which the length-dependent forces of opposing muscles generates equal and opposite torque about the joint. Postures are equilibrium configurations that "attract" different body segments. This view of posture has been more recently extended to the analysis of movement and trajectory formation: as the distribution of neuromuscular activity varies in time, a sequence of attractors is defined (virtual positions) therefore generating a movement and/or modulating a contact force. This approach to motor control is readily applicable to artificial systems by characterizing the actuators (together with their local controllers) as elastic elements with selectable stiffness (or compliance) properties.

In previous papers (Marino et al 1985, 1986, Morasso and Mussa Ivaldi 1986, Morasso et al 1987, 1988, Mussa Ivaldi et al 1988) we have shown that muscle elastic properties lead to a natural representation and regularization of redundancy. In essence, redundancy resolution derives from the simulation of an externally imposed motion (for example a displacement of the end-effector) which represents the selected task. Given such a perturbation, a physical

system characterized by elastic actuators will settle to a well defined configuration which corresponds to a minimum of potential energy compatibly with the externally-imposed motion. The generation of a full trajectory is derived by iterating the above process with small (infinitesimal) position changes along the desired path. At the end of each displacement, the control inputs are reset so as to maintain the system at equilibrium: This regularization process has been called Passive Motion Paradigm (PMP). In contrast with conventional approaches to redundancy (e.g. the application of standard jacobian pseudo-inverses or the optimization of motions in the null-space) PMP leads to a class of integrable local solutions which can be adapted to the specific context by the appropriate choice of actuator elastic properties.

3. Abstract Force Fields

According to the principle illustrated above, planning a human movement, i.e. generating the synergy of muscle activations, is an internal simulation of passive movement, or a motor "relaxation" between several different postures.

However, the power of this principle is that we can abstract from the specific physical nature of the underlying process (the viscous elastic properties of the actuators). Even if the actuators do not have these properties, we can still attribute a virtual compliance to them, which can have a natural interpretation in any specific planning circumstance.

The general concept is that we can solve motor redundancy by means of a multi-dimensional force field which can model directly a physical substrate or can be an abstract representation: an abstract force field (AFF).

From the motor P&C point of view, *attractor postures* are representations that can be programmed incrementally (with respect to the *current posture*) for obtaining two complementary results in the configuration space. The generation of prescribed trajectories (if the motion is free) or of prescribed interaction forces (if the intended movement is inhibited by external constraints). The relationship between the two entities is given by the *stiffness matrix*:

$$\tau = K \delta q$$

where δq represents a small displacement from an attractor posture, τ is the corresponding restoring force/torque vector and K is the stiffness matrix. The generality of this formulation is that we can build upon it different representational layers that capture diverse aspects of motor P&C in the same framework, by superimposing (in space and time) several AFFs. For example, the constraints due to joint limits can be easily taken into account by an AFF where repulsive forces steeply increase the potential function in the neighborhood of forbidden joint configuration regions. The problems that plague force control (e.g. instability) are overcome because interaction forces are not controlled explicitly but arise implicitly from the intrinsic controllable stiffness/compliance of the system.

Other P&C issues are better formulated in 3D space, for example trajectory formation or obstacle avoidance, and this implies a mapping of the AFF between configuration space and 3D space. This mapping must satisfy the principle of invariance of virtual work:

$$\delta q \cdot \tau = \delta x \cdot f$$

where δq and δx are virtual displacements in the two spaces and τ and f are the corresponding force vectors. From this it is possible, in principle, to determine the passive behaviour of the system when a disturbing force f or a passive displacement δx is applied: δq identifies indeed the resulting dislocation from the attractor posture and τ is the corresponding restoring force vector.

The power of the method is threefold: We reduce free movements and interaction movements to the same paradigm, we overcome the redundancy issue (planned trajectories are guided in the direction of minimum potential energy), and we open the window to a variety of generalizations of the $\langle f | \delta x \rangle$ representation. At the basic level, $\langle f | \delta x \rangle$ specifies an attractor that implements the inverse kinematic problem of a redundant kinematic chain. More general is the case of a whole humanoid, composed of a set of (possibly closed) kinematic chains: This requires to segment the $\langle f | \delta x \rangle$ representation in order to specify an attractor for each limb of the humanoid. Moreover, obstacle avoidance can be expressed as a repulsive force field emanating from the obstacles, overlapped with the attracting field associated with the target.

The technique of overlapping multiple AFF's (some of them intended to attract and some to repel the motion of the robot) can be extended to the time axis, allowing the smooth composition of complex trajectories driven by discrete instantiation of guiding postures/spatial patterns, in a similar way to the segmentation of continuous speech in overlapped phonemes or of cursive script handwriting in overlapped graphemes. Furthermore, the technique can be used for both off-line planning and on-line correction.

Common Sense provides a conceptual methodology. Complex Dynamics is better handled if data are explicitly divided from processes, i.e. equations. In this sense we can draw an analogy between the Abstract Data concept in Software Engineering and the AFF concept considered in this paper.

4. Hybrid Motor Schemas

Hybrid motor schemas (HS) are intended to capture the complexity of actions in the real world. The term "hybrid" refers to the integration of Logic/Symbolic and Analogic/Simulative representations. Let us consider "grasping", for example, performed by a dexterous hand. Its structure can be detailed from different points of view:

1. There are different grasping modes;
2. The shape of objects and their physical nature provide several virtual "handles";
3. The layout of the working environment provides constraints on the approach phase and the selection phase.

Grasping modes are general strategies for grasping, regardless to the particular points ("handles") on which the contact will be realized: *two-fingers*, *palm*, *cylindric*, *pencil-type* and so on.

Handles are the possible known points for stable grasping, defining families related to grasping modes by higher level semantic functions.

Finally, grasp planning must put into relation the actual position, the position/orientation and size of the object, the selected grasping mode and handles, the working environment and its constraints, giving as an output a trajectory in the n -dimensional space of the joints.

A grasping schema needs analogic representations for the different grasping modes, parameterized in such a way to adapt them to the specific dimensions of the handles. It also requires rules for selecting grasping modes and parameters according to the context, and procedural attachments for activating the motor tasks.

While the purpose of activating schemas is usually that of directly producing inferences, the purpose of a hybrid motor schema, such as the grasping schema, is to set up an AFF whose simulation details the plan of movement. The coordination among all the fingers as a function of the specific shape of the grasped object, for example, is not explicitly represented in the schema but is an indirect result of simulating the complex dynamics of the force field. The different phases of the planned actions (e.g. approach and grasp) can be joined together by singling out a limited set of "key postures" (Morasso, Vercelli, Zaccaria, 1988). In summary, a hybrid motor schema is a representation of the whole movement or a particular chunk of it which is conceptually divided into two parts:

- *an analogic part*, i.e. kinematic and/or dynamic and/or eidetic descriptions;
- *a symbolic part*, i.e. micro and/or macro features that specify something of the movement.

The *analogic part* of the Grasping Schema is basically composed of two parts: the 3D-iconic representation of the scene (start and end positions of the hand, obstacles), and processes: those which adapt the grasping mode to the particular object, those which set up the AFFs connected to target and obstacles, and the AFFs that drive the kinematics of the whole structure.

Differently from classic planning systems, the behavior of the system is the result of many cooperating processes. In particular, the time is not explicitly described in the HS. The temporal performance is "observed" while the grasping action is being carried out, driven by the composition of the AFFs. However, the temporal behavior can be changed by selecting appropriate values, for the fields strengths, for the stiffness matrices, and so on.

A more detailed analogic model for a redundant anthropomorphic robot based on PMP and connectionist concepts will be presented in section 6.

5. Using AFFs and HSs: Some Examples

Translating the AFF and HS concepts into practical computational terms is our long-term goal. We have just begun to setup a few tiles of the mosaic, developing software tools for various simulation purposes. One such tool is a prototype of a human grasping movements planner, called OCTOPUS. This system is able to solve reaching-grasping problems, starting from a set of icons (3-D kinematic skeletons) of the hand and from a hybrid schema of the action.

Icons are organized as a Knowledge Base. They represent the *analogic part* of the planner. The planner itself performs symbolic computation, or *reasoning*, on icons' properties, also described in the KB.

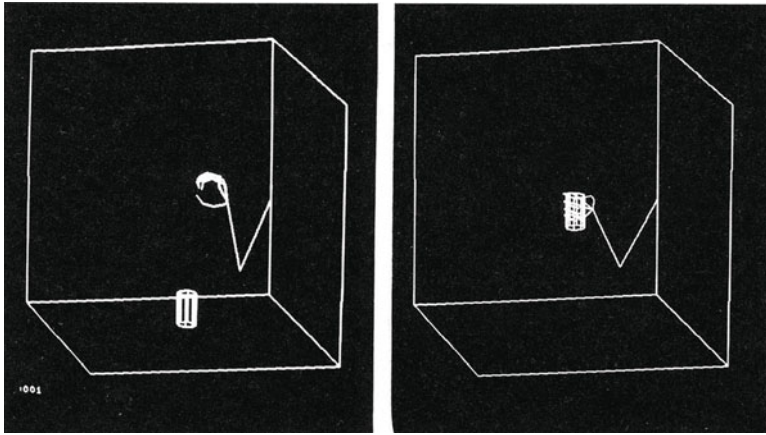


Figure 1 - *arm reaching task*, initial and final positions

An icon is usually a virtual target for a part of a movement, as a final or intermediate posture; in some case, an icon can be a *repulsing* posture, having the meaning of a forbidden (e.g. dangerous) configuration. Several icons may be (and usually are) simultaneously targets, so that no one will be exactly part of the actual motion. The aim of the planner is to:

1. choose the proper icons (related to the the particular object, and the given grasping type), e.g. a suitable grasping posture ;
2. *adapt* the icons to the actual problem (changing parameters so that they correspond to the shapes of objects); e.g. the grasping posture is made fit to the object;
3. set the appropriate Force Fields associated to the set of icons (which may include obstacles, constraints etc.).

The planner carries out its task on the symbolic "summaries" of the icons, also stored in the KB. After the planner has decided and set up the *analogic model* for the action, a *realizing process* starts, which generates the movement. This may be done using the P&C paradigm described in the previous section. Currently, computation is carried out by procedures written in NEM language. The *NEM language* (Marino et al, 1985,1986) allows building arbitrary structures of geometric frames, to compute geometric relations among them, and to specify relative motions in a concurrent way. The KB, in this example, is a *multiple inheritance semantic network* of *frames* implemented in Lisp; frames are either *physical objects* or *processes*. An *action* (in our case, a grasping action) is a compound process which connects the planner to the actual object, computes the attracting icons (e.g., the "handles" of the object), then sets up the AFFs and supervises the different phases. Analogic procedures are *procedural attachments* to concepts in the hybrid KB (Adorni et al. 1988). Figures 1-2 are simulation outputs of the planner: the reaching task is performed by an inverse kinematic module that is the HS "arm reaching task", while grasping task is performed by an interpolation module that is the HS "handle grasping task".

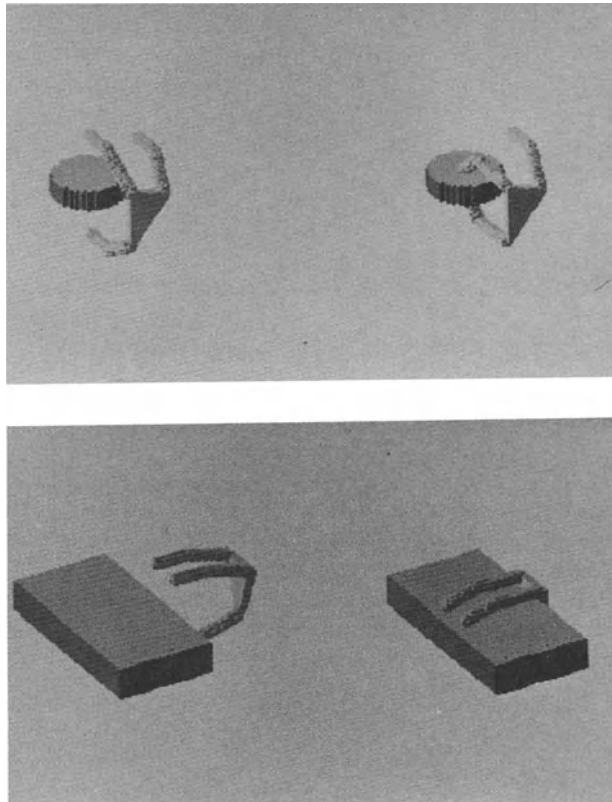


Figure 2 - *handle grasping task*, actions for different objects

For the organization of such HS's we started building a tool for *body models*, that encodes the special geometric constraints of degrees of freedom, a tool for *compliant actuators*, that sets up the graph implementing the attractor posture, and a tool for *obstacle modelling* that implements the repulsive force fields as AFF. The figures show some examples of simulations performed within such a software environment. Figure 3a is an example of inverse kinematics for a redundant arm (open kinematic chain). Figure 3b extends the example to the parallel movement of two arms towards to independent targets. Figure 3c shows the generation of a whole body movement from two subsequent attractor postures.

The examples in figures 4 and 5 lie outside the anthropomorphic scenario. They show the expressive power of the AFFs approach, regardless of the type of the robot, when dealing with an inherently redundant / partially undefined task. The problem of figure 4 is a pick-and-place task with obstacle avoidance, using a 5-d.o.f (not including gripper) robot arm. Grasping modes are stored in the HS, and they are *attractor icons* together with the final ("place") position, whereas obstacles are *repulsive icons*. The whole movement is a simultaneous evolution of all d.o.f. in a global AFF set by the planner.

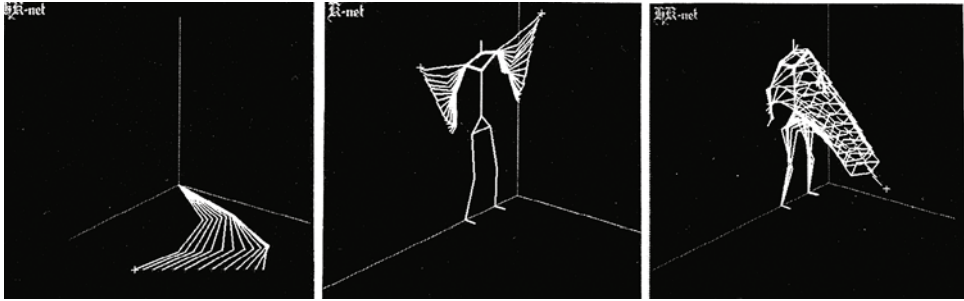


Figure 3 a,b,c - Movements generated by HSs and AFFs:
a) redundant arm, b) independent targets, c) via *attractor postures*

The final problem is planning trajectories for a mobile, conventional robot for accomplishing continuous arc welding tasks. Such task involves 5 d.o.f., whereas the robot has 6 d.o.f., and additional d.o.f. are typically provided in some other way. Figure 5, for example, shows two phases of a common task in which two small pipes are to be welded to a bigger one (horizontal); the big pipe rotates along its axis and the robot can move along a line parallel to that axis, hence 8 d.o.f. are available. All d.o.f. must be used to reach the object to be welded, and to follow a "big" trajectory (in our example, the intersection of two pipes of relevant dimensions) avoiding, at the same time, collisions and illegal postures, constrained by the welding hardware.

This problem has been solved for a real robot using an AFF technique (Casalino et al. 1988) similar to that described above, in which the relevant icons were guiding the robot movement among the different phases, and "pushing away" the arm from illegal or colliding postures.

6. M-nets

A modelling framework, called "Motor Relaxation Networks" (shortly, M-nets), has been formulated to express motor planning for the biological motor control system (Morasso et al., 1989). This method, which is based on Passive Motion Paradigm (Mussa Ivaldi et al., 1988), can be formulated in connectionist terms as a multiple constraints satisfaction network, driven by a potential function, similarly to Hopfield networks (Hopfield 1982) because it is possible to define a global function E , which has the property of a potential function. In Hopfield nets E is a purely abstract computational concept; in M-nets, on the contrary, E is the elastic energy stored by the muscles, which is the direct consequence of a physical phenomenon of biological significance. In biological motor control, a sufficiently realistic M-net may be used as a tool to formulate predictions of the structure of complex motor synergies. Therefore, M-nets are a kind of analogical model in a double sense: because the computational process is a dynamic system and because there is a direct analogy between the structure of the model and the underlying physical world.

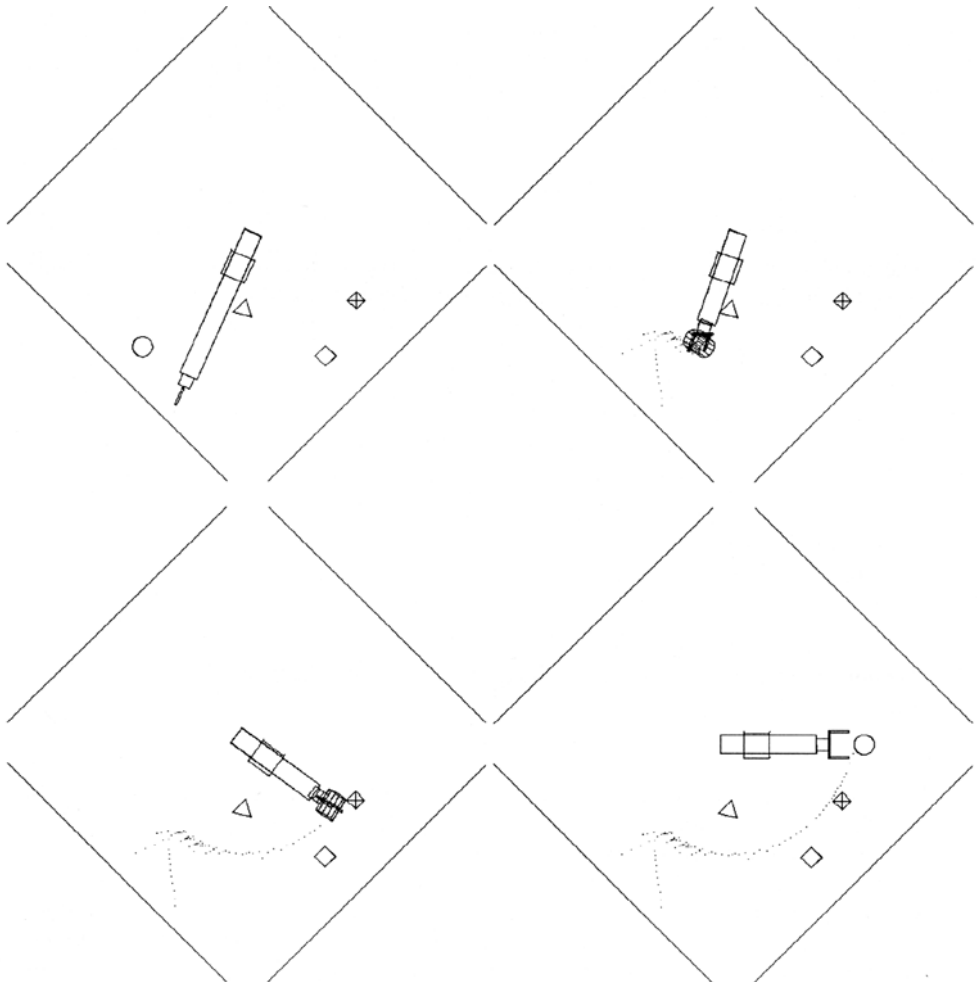


Figure 4 - Top view of a pick-and-place movement via AFFs

Three types of units are present in M-nets:

- skeletal segment units (s-units),
- muscle units (m-units),
- ligament units (l-units).

Let us now consider concisely their structure and function:

1. **S-units** model the different skeletal body segments, represented as rigid bodies to which complex sets of forces (muscle forces, ligament forces, external forces) are applied.

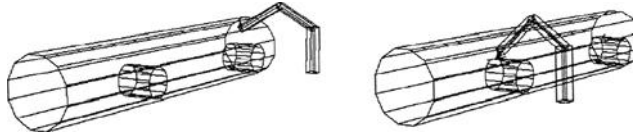


Figure 5 - An 8 d.o.f. task: continuous arc welding of big pipes

These units are more complex than those usually considered in connectionist models because they deal with vector quantities (force, velocity, and rotation vectors) and perform vector operations such as vector products. The correct analogy, for an s-unit, is with a whole cortical column, not with a neuron.

The input is a set of force vectors ($f_1, f_2, \dots, f_n$) applied to a set of insertion points ($p_1, p_2, \dots, p_n$) distributed over the skeletal segment.

The outputs are the positions of the same insertion points.

The transformation operated by an s-unit is characterized by the following steps:

- compute the resultant force and torque vectors (F and N),
i.e. the net vectorial input;
- compute incremental displacement and rotation vectors as
sigmoidal functions of F and N , respectively;
- update the insertion points with a rigid translation/rotation.

2. **M-units** model the muscles as viscous-elastic elements characterized by a family of curves

$$f = f(l, \dot{l}; \alpha)$$

indexed by the level of muscle activity α (f is the muscle force, l is the muscle length and $\dot{l}$ is the speed of contraction)¹. M-units are connected to two s-units in the case of single-joint muscles and to several s-units in the case of multi-joint muscles (e.g. many muscles of the hand). The inputs are the insertion points of the muscle, from which the muscle length is computed. Two kinds of outputs are present: one is muscle force, which is transmitted to the connected s-units, the other is muscle activation α that represents the motor output and should be transmitted to the muscles. During the simulation of passive movements, muscle lengthening/shortening is accompanied by variations of α that keep the force output approximately constant.

3. **L-units** model the joints as elastic connections with high stiffness. Complex joints can be modelled with multiple l-units, even joints distributed over several skeletal segments, such as the wrist joint. Another function of l-units is to model joint limits in a soft way, by generating increasingly stronger restoring forces as joints approach forbidden configurations. Note that joints and joint limits are not represented as hard constraints

1. We used different kinds of models for $f()$, derived from the literature (Hatzee, 1981).

but as soft constraints. The power of this concept is that it brings geometry, kinematics, and actuator dynamics under a uniform representational scheme. From the operational point of view, l-units are simplified versions of m-units, without the mechanism of modification of α . L-units and m-units can store elastic potential energy.

M-units and l-units behave as impedances, i.e. they receive positional information and react feeding back force information. S-units, on the contrary, behave as admittances, i.e. they receive force information and react modifying positional parameters. In relation with usual connectionist models, m-units and l-units are analogous to connection weights, these weights having the meaning of stiffnesses: fixed in the case of ligaments and variable in the case of muscles.

A specialized s-unit is the "environment" (or "ground"). Gravity is represented by fictitious l-units that link to ground all the regular s-units. Environmental constraints that do not allow the movements of some skeletal segments also can be represented by means of fictitious l-units as well as motor tasks, which can be specified by means of fictitious l-units that link goal-oriented s-units (e.g. a distal phalanx in a reaching task) to the environment. Concurrent tasks are represented in a natural way by activating simultaneously multiple fictitious l-units.

An M-net is an asynchronous network. At equilibrium, the set of forces that enter each s-unit give resultant force and torque vectors which must be null: as a consequence, the positions of the insertion points of all the muscles and ligaments remain stationary. However, even when the network is at equilibrium, muscles and ligaments may well be far from equilibrium and may in fact store considerable amounts of elastic energy.

M-nets solve the inverse kinematic problem both at the joint level and at the muscle level, irrespectively of the degree of redundancy. On the contrary, most of the methods developed in robotics for dealing with redundant systems (explicit optimization and/or motions in the null-space) are only practical for simple kinematic chains and are not scalable up to the degree of complexity of the human musculo-skeletal system. Moreover, M-nets do not solve inverse kinematics only. The solution, which consists of the parallel modulation of elastic actuators, has built-in compensatory characteristics that may counteract the forces due to dynamics. At least for simple arm movements (Flash 1987) it has been shown that this kind of dynamics compensation mechanism is sufficient to explain the experimentally recorded deviations from the straight line.

We think that the major source of flexibility of M-nets with respect to the other connectionist models of motor control (Bullock and Grossberg 1988, Jordan 1988, Kawato et al 1987, Kuperstein 1987) is that M-nets explicitly use the viscous-elastic properties of muscles. A unit in M-nets plays the role of an entire cortical column in motor cortex, and the network is the modelling framework to express motor planning and control of highly redundant robots, actuated by means of muscle-like devices.

A preliminary version of an M-net simulator was implemented by modifying a motor simulation and animation system - NEM - previously developed (Marino, et al., 1985). The simulator allows the definition of arbitrary complex musculo-skeletal systems in three dimensions, and a special graphic interface allows 3D animation of the simulated tasks on a graphic work-station. The output of a simulated movement for an anthropomorphic finger is shown in figure 6.

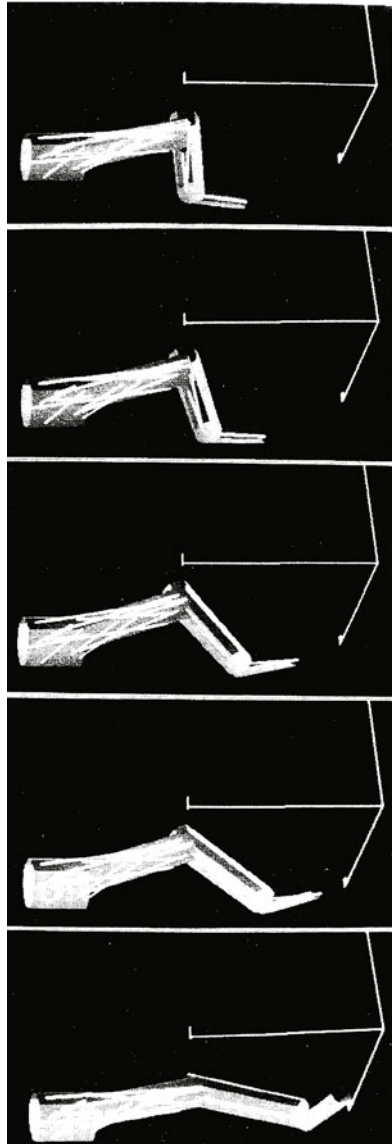


Figure 6 - Movement of an anthropomorphic finger using M-nets

Although M-nets for complex musculo-skeletal structures run presently far from real-time conditions, they could quickly gain from modern parallel computing techniques. In particular, implementation on a transputer architecture seems to be feasible and natural and this is actually our next research target.

7. Analogic Planning of Action Sequences

So far, we have been concerned basically with generation of laws of movements (AFFs, HSs), and on following trajectories with complex kinematic chains (PMP). Now let us face the problem of finding the temporal relationships between motions for solving a goal. In other words, the intrinsic constraints of a task must be detected, giving the *sequence* of events necessary to accomplish it. This is the classic field of application of Planning in A.I.

In this section some new concepts about planning will be outlined. We start from the ideas introduced above, namely, the definition of AFFs, HSs, and PMP. Our aim is to show how action planning may be formulated in terms of movements inside *Force Fields*, producing an *analogic or hybrid model*.

According to the PMP concept, we can realistically restrict ourselves to the formation of trajectories for each part to be handled, as if they could *fly* in space. Once the "flight trajectory" of each part is determined, together with their time order, a hand can be *passively* guided by it to solve the inverse kinematic/dynamic problem using PMP.

A 2-D Block World problem will be used as an example of a real (more complex!) *assembly problem*.. However, our world has some "unusual" (for classic A.I. formulations) additional features, like *shape*, *stable postures*, and *concurrency*. Such features are the source of complexity in the purely symbolic approach, whereas they are the main source of information in the analogic scheme.

Let us consider a set of parts that must be assembled into a whole. In fig. 7a a cube, an open box and a cover are randomly placed on a table. It is in general possible that some piece prevents the movements of some other, as in our case, where the cover is in origin placed on top of the cube.

In general, the final structure of the assembly is described by a set of phrases in a suitable language, at the *symbolic* level. Such language is based on formulas that define the geometric relationships between (at least) couples of parts. Each formula is a *relation*. Besides the purely geometric quantities there must be some other attributes of the relation which specify the *semantics* of it: typically, if it is *fixed* or *free*, or, in general, the degrees of freedom left and their dynamics.

In A.I. and Robotics languages for specifying assemblies have been investigated since several years (see for example (Lieberman and Wesley 1977)); however, they are used in a purely logic/symbolic paradigm. In the example of figure 7, a general relation specifying the final position of the cube (fig. 7c) may look like

rel cube1.face3 on box.face2 <transf> free

rel box.face2 on table <transf> fixed

where *transf* is the geometric relationship between the intrinsic reference frames of *cube1.face3* and *box.face2*. A similar relation can be written for the cover and the box:

rel cover.face2 on box.face4 <transf> free

where, for example, *box.face4* is the small top of the "left arm" of the U-shaped box.

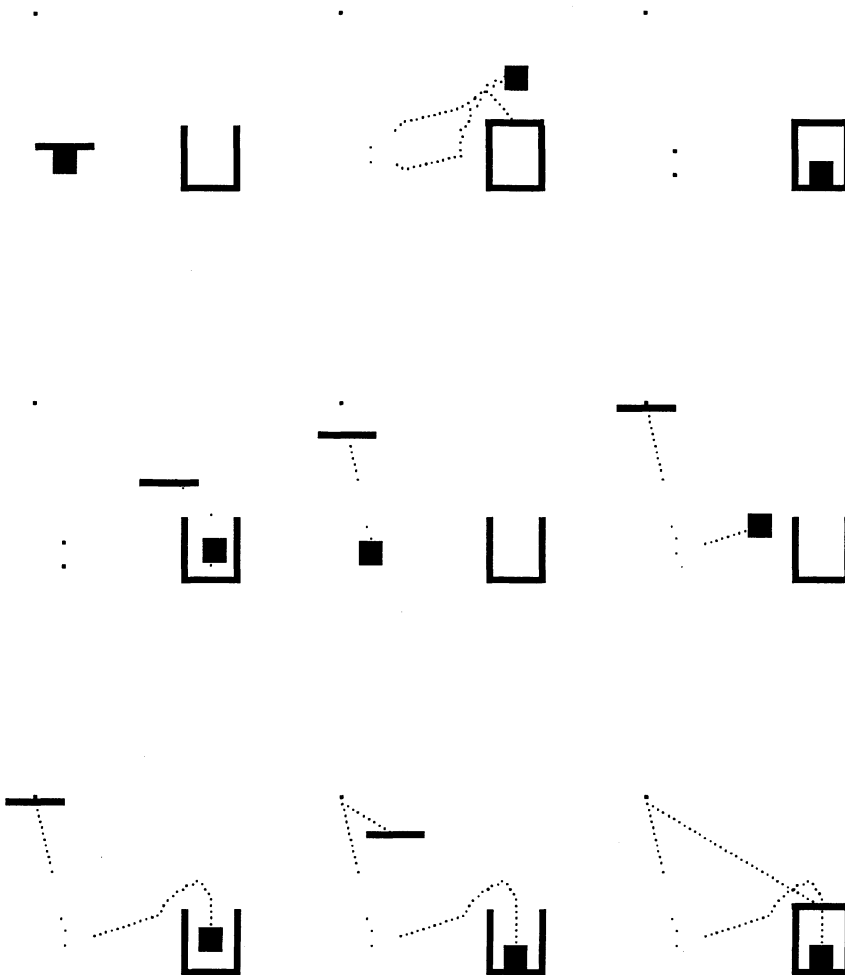


Figure 7 a-i - Analogic planning of the *cube-in-the-box* problem

Many human - oriented instruction sheets for assembly communicate the reader a comparable information. The table in figure 8, showing an "exploded" vacuum cleaner, shows basically local relations, leaving the reader the task of deciding what to do first. A model of this complexity should be a good testbed for the algorithms shown below. Note, by the way, the intuitive evidence of how the concept of "exploding the assembly" gives information about the sub-actions sequence (this concept will be used later on).

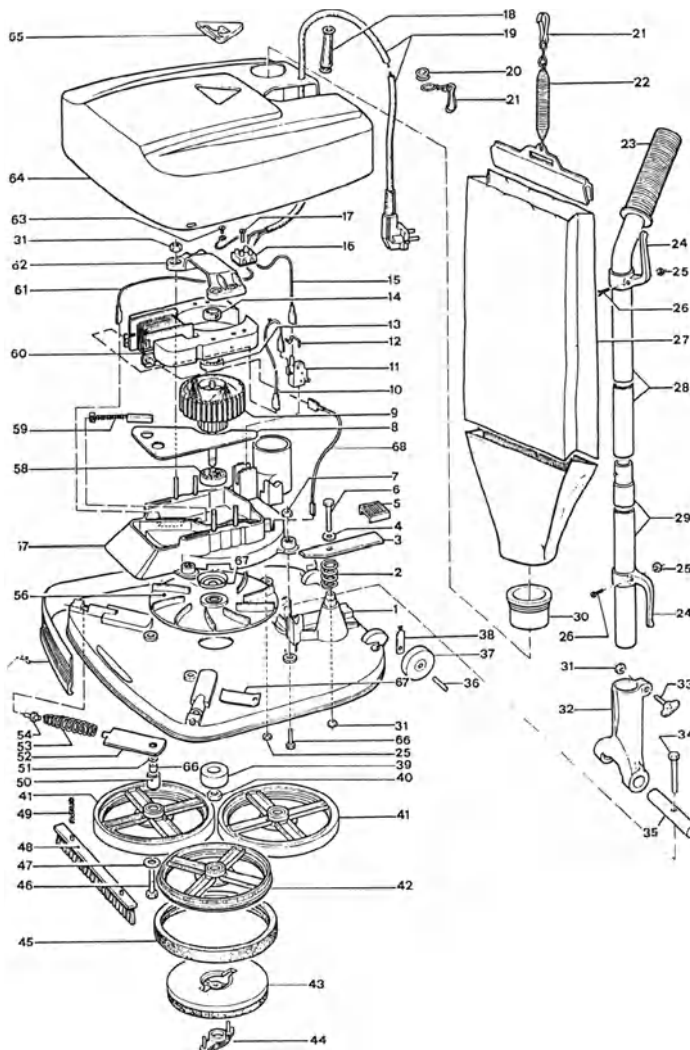


Figure 8 - A typical human-level assembly sheet

7.1 Measurements vs. Inferences

In general, relations define *local* information about the final assembly. For example, the relation between the cube and the cover, say it R_{CV} , is not explicitly present in the previous set of relations. The traditional approach is to define a formal logic model and to prove theorems inside it. So, the relation R_{CV} can be deduced by the set of assertions and inference rules. Another possible approach is to build an *analog model* (Funt 1983, Johnson-Laird 1983) of the final assembly, and to perform measurements on it. Such model may be physically

analogic (for example, a 3-D memory), or some kind of simulation structure. Once done, all global properties of the object can be discovered by direct measurements (distances, relations in general) on the "mental model". There are some important relations, from the point of view of planning, that must be found: they will be described in next sections.

Similar relations must be defined for the non movable parts of the environment. We omit, for sake of briefness, a discussion on how to define *shapes*, assuming available some general formal language like in (Popplestone and Ambler 1978).

7.2 Building the Mental Model

This is the first phase of planning. It consists of putting together all the partial relations and of building the whole assembly model; generally speaking, it contains more explicit information than the original set of partial relations.

Starting from the initial position (which can be described by a similar set of relations), the final model can be analogically built with a *relaxation model*, a sort of *spatial Hopfield network*, as follows:

1. For each couple of pieces inside a relation, connect them by a *spring* (see below).
2. Make each piece generate a repulsive force field to every other piece.
3. Consider each piece as movable, subject only to the forces generated by springs and by repulsions.
4. Let all pieces move simultaneously in space, with dynamics suitably defined so that viscous parameters dominate the motion's law, hence avoiding oscillatory movements.
5. When a local target relation is reached, turn on other external forces defined in the environment (e.g., gravity) acting on the involved pieces.
6. Monitor continuously the local and global potential energy stored in the springs, that should reduce to zero once all relations are satisfied (by the way, the model can converge also in presence of non consistent sets of relations, by simply not reducing to zero the energy).
7. Start a *backtracking action* when some local minimum is reached; local minima are equilibrium positions, or deadlocks, in which two or more pieces want to move toward their goals, disturbing each other. Two examples are: i) while navigating toward a direction, you enter into a concavity of an obstacle or of group of obstacles, ii) while the cube tries to enter into the box, the cover is already in its final position. Deadlocks require to climb back the gradient of the global energy field.
8. Start a *Backtracking Action* when some non stable situation occurs: e.g., a piece falls down due to external forces. Consider it a deadlock too.
9. Mark the deadlocks for the successive planning phase.

This phase can also be considered as the analogic model of an assembly phase with an infinite number of hands available.

7.2.1 Springs

Springs are defined so that a roto-translatory force is generated by a law of the form (for the planar case)

$$\begin{cases} F = K_l \Delta l \\ M = K_a \Delta \theta \end{cases}$$

F and M are the force and torque generated by the misplacing of the piece with respect to its attracting posture. In practice, a saturation on the force/length curve is required in order to avoid too high accelerations at the starting positions.

7.2.2 Deadlocks

The possible deadlock situations are important. They are *intrinsic sources of partial ordering among the actions*, and the planner must recognize them. We may use the term *rendez-vous* (RV) to denote the instant in which two parts meet to form a subassembly.

There are four basic situations.

1. *Dead end*. A piece, navigating toward a target, is in a local minimum of the field. This does not involve a different partial ordering per se, but can be solved introducing a (dummy) obstacle in the point where the deadlock occurred.
2. *Intersecting Work Spaces*. A subassembly lies one half on one side of another subassembly, one half on the other side. An example is a bolt that crosses a plane surface, with a nut on the other side: the RV of the subassembly bolt/nut must be delayed after the RV between the bolt and the surface.
3. *Box*. An opened cavity (possibly infinite) cannot be closed until the contained subassembly reaches its minimum energy (see figs. 7b and 7g-h-i).
4. *Instability*. The external forces disassemble an already obtained subassembly. For example, the top of a table cannot be put on only two legs; such RV must be delayed.

Discovering deadlocks can be seen as a form of *temporal measurement* (the analogic counterpart of *temporal reasoning*) on the analog model. Many of them can be checked in the first phase, possibly by executing the process several times with random laws of motion. Other deadlocks must be discovered in phase two.

7.3 Disassembling for Ordering Actions

The mental model of the assembly can now be disassembled (exploded) by:

1. eliminating the springs,
2. inserting new springs between pieces and their starting points
3. making each object repulse each other.

Disassembling an object is a form of measurement that gives information on partial orderings. By monitoring the relative speeds of pieces leaving their positions, it is possible to devise which piece can move first; pieces free to move first during the explosion are candidate for being the last parts to be joined together. For example (fig. 7d), the cover goes away rapidly, while the box is slower, indicating that the RV cover-box must follow the RV cube-box. A similar situation occurs for the bolt-plane-nut, whereas a different technique must be applied to the "instability" case.

If the assembly is composed by many parts, the explosion phase can be iterated for the inner parts to get the desired resolution in the monitoring process.

By disassembling the object, a List of Partial Orderings (POL) can be built.

7.4 Third Phase

The third and last phase involves checking the correctness of the POL with respect to the initial situation. It may happen, in facts, that moving the piece P1 before P2 generates a deadlock; for example, if P1 is supporting P2. This can be checked by the following procedure:

1. select each piece to be moved, one at the time, following the POL;
2. connect it by a spring with its final position;
3. let all the other pieces be movable obstacles;
4. let the selected piece go to its target place;
5. monitor the movements of the others, finding relevant deadlocks;
6. for every deadlock, generate a movement able to avoid the deadlock, and put it into the POL *before* the selected action.

For example, the cube is under the cover. By moving the cube, the cover is pushed away; this requires displacing the cover to some free position, or to grasp it (if we have a free hand to keep it while another hand acts on the cube). In general, in this phase the original setup of parts is considered an assembly S0 that must be disassembled to compose a new one, S1. If not every piece is free to go from S0 to S1 following the POL, S0 must be (partially) disassembled. Such process requires the same analog modelling technique as before, plus a set of possible *free places* where to put down the single parts, if necessary.

At this point, we have the complete POL, which is a series / parallel ordering of actions.

7.5 Concurrency vs. Sequence

The RV constraints found so far are intrinsic. Therefore, many parts may still be moved concurrently. If we have a reduced number of hands, concurrent operations may require a serialization, possibly total. In some cases this is not possible: for example, a bolt cannot be left inside a vertical hole, with its head down, waiting for the nut. These situations can nevertheless be checked using the same analog model.

As an example, in figure 7 the assembly of *cube in the closed box* is shown, in the classic *block world* paradigm, typical of A.I. (serial action, one hand). Fig. 7a is the start position, fig. 7c the final. Phase 1 has found a deadlock in fig. 7b; fig. 7d is the "explosion" of the mental model; fig. 7e is the beginning of phase 3, when the cover is pushed away when attempting to move the cube. Figs. 7f through 7i show the actual assembly phase: the cover has been displaced, then the cube is moved to its final position, then the cover is placed on the box.

The POL, in our case, is obviously

((cover *FreePoint1*) (cube *CupBottom*) (cover *CupTop*))

A comparison with a traditional STRIPS planner solution, for example, is left to the reader.

The solutions shown in the previous figures were found by a preliminary implementation of the Analogic Planner, written in **Nem++**, a language for spatial and temporal models which is an engineered development of **Nem** (Marino et al. 1985).

Acknowledgements

This work is partially supported by EEC **FIRST** Project and by **Special Project on Robotics** of the Italian Research Council.

References

- Liebermann, L.I., Wesley, M.A. (1977) AUTOPASS: an automatic programming system for computer controlled mechanical assembly, IBM J. of R. & D., July 1977
- Popplestone, R.J., Ambler, A., and Bellos, I. (1978): RAPT: a language for describing assemblies, *Industrial Robot*, vol. 5 no. 3, 131-137
- Hatze, H. (1981) Myocybernetic control models of skeletal muscle, Univ. of South Africa, Pretoria
- Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities, *Proceedings National Academy of Sciences USA*, 79, 2554-2558
- Funt B.V. (1983) Analogical Models of Reasoning and Process Modeling, *IEEE Computer*, 16:10, 99-104
- Johnson-Laird P.N. (1983) *Mental models*, Cambridge University Press, Cambridge
- Marino G., Morasso P. and Zaccaria R. (1985) Motor Knowledge Representation, Proc. IJCAI (Los Angeles, California)
- Marino G., Morasso P. and Zaccaria R. (1986) NEM: a Language for the Representation of Motor Knowledge, in *Human Movement Understanding* (P. Morasso and V. Tagliasco Eds.), North Holland
- Morasso P. and Mussa Ivaldi F.A. (1986) The role of physical constraints in natural and artificial manipulation, Proc. IEEE Conf. Robotics and Automation (San Francisco, California)
- Morasso P., Mussa Ivaldi F.A. and Zaccaria R. (1987) Redundant robotic manipulators. I: regularizing by mechanical impedance, *NATO ARW Kinematic and Dynamic Issues in Sensor Based Control*

- Flash, T. (1987) The control of hand equilibrium trajectories in multi-joint arm movements, *Biological Cybernetics*, 57, 257-274
- Kawato, M., Furukawa, K., and Suzuki, R. (1987) A hierarchical neural-network model for control and learning of voluntary movement, *Biological Cybernetics*, 57, 169-185
- Kuperstein, M. (1987) Adaptive visual-motor coordination in multijoint robots using parallel architecture, *IEEE Conf. Robotics and Automation*, 1595-1602
- Mussa Ivaldi, F.A., Morasso, P., and Zaccaria, R. (1988) Kinematic networks - a distributed model for representing and regularizing motor redundancy, *Biological Cybernetics*, 60, 1-16
- Adorni G., Camurri A., Poggi A., Zaccaria R., (1988) Integrating Spatio-Temporal Knowledge: a hybrid approach, 8th European Conf. on Artificial Intelligence, Munich, GFR
- Casalino G., Vercelli G. Zaccaria R., (1988) Trajectory formation and path planning of robotized continuous welding tasks, 5th ESPRIT European Conf., Brussels
- Morasso P., Vercelli G., and Zaccaria R. (1988) Rappresentazione del movimento in Robotica: modello, ricostruzione, data base cognitivo in un caso antropomorfo, 4-th SIRI National Conf. (Milano, Italy)
- Steels L., (1988) Steps Towards Common Sense, proc. of ECCAI88 Conf. (Munich, GFR)
- Bullock, D., and Grossberg, S. (1988) Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation, *Psychological Review*, 95, 49-90
- Jordan, M. (1988) Supervised learning and systems with excess degrees of freedom, COINS Technical Report 88-27
- Morasso, P., Mussa Ivaldi, F.A., Vercelli, G., and Zaccaria, R. (1989) A connectionist formulation of motor planning, in *Connectionism in perspective*, Pfeifer R., Schreter Z., Fogelman F., and Steels L. Eds., Elsevier

Structural Constraints And Computational Problems in Motor Control.

F. A. Mussa-Ivaldi and E. Bizzi
Massachusetts Institute of Technology,
Cambridge, MA 02139,
USA.

1. Introduction

A central dogma of computational approaches to motor control is that in order to understand the biological organization of information processing it is crucial to understand the nature of the problems being solved by the motor system and of their possible solutions. Such a statement has defined a common ground for investigations in robotics and neuroscience. Among the computational problems considered in robotics, a prominent role has been occupied by finding the time course of joint configurations given a desired motion expressed in task-coordinates (inverse kinematics) and finding the torques that must be applied to achieve a predefined motion of a mechanism (inverse dynamics).

But, if these are fundamental issues in motor control then why have so few insights in robotics proven to be of some biological relevance? One answer to this question comes from the observation that the structural (i.e. mechanical and geometrical) properties of motor systems contribute significantly to the definition of the computational problems and to their solutions. For example, in several robotics applications the actuators are force generators with negligible impedance properties. Such a "low-level" actuator property is sufficient to motivate extensive research in problems of inverse dynamics, whose goal is the specification of force patterns. Another feature characterizing many artificial manipulators is their lack of redundancy. Only six kinematic parameters are required to completely determine the location of a rigid tool in the workspace. Consequently, the most "standard" manipulators have been designed so as to have six independently controlled degrees of freedom. In this case, inverse kinematics is a well-posed problem which can be solved by inverting the manipulator Jacobian matrix. Therefore, for a non-redundant manipulator the problems of dynamics and control can be addressed independently of kinematic issues. In fact, the latter have unique solutions for any given task.

This is clearly not the case with biological systems which are instead characterized by a large degree of kinematic redundancy. For example, we can keep our hand in a given location of the workspace with different sets of joint angles. Given a desired location of the hand, the choice of a body configuration has a direct influence on the hand impedance. The hand impedance characterizes the force generated in response to an externally imposed motion. Thus, any particular kinematic solution has an immediate effect on the control of the interactions with the environment. Another substantial difference between artificial and biological systems regards the property of the

actuators. As we stated above, robots are typically operated by nearly ideal effort sources: sources whose force output does depend upon position. In contrast, the biological muscle makes a "poor" actuator when considered as a force generator. A muscle is indeed characterized by non-negligible elastic properties which depend upon its level of activation.

In this paper we will start by reviewing some experimental studies which demonstrated that these "tunable" elastic properties of muscles are not a disadvantage for the neural controller. Quite on the contrary, they provide the necessary physical substrate for controlling posture, movement and contacts with the environment in a unified way and, ultimately for avoiding inverse dynamics computations. Then, we will show how the elastic properties of the muscles lead to the regularization of the ill-posed problems associated with kinematic redundancy.

2. Actuator mechanics and computation

Several investigators of biological systems (Feldman, 1966, Rack and Westbury, 1969, Bizzi et al., 1976, Nichols and Houk, 1976) have focused on the role of muscle mechanical properties in motor control. The force/length relationship at steady-state in individual muscle fibers was studied by Gordon et al. (1966). These authors related the development of tension at different muscle lengths to the degree of overlap between actin and myosin filaments. (In the structural organization of vertebrate striated muscles sarcomeres are the constituent units which are repeated longitudinally along the muscle fibers. These units consist primarily of comb-like arrays of overlapping actin and myosin filaments.) The process of generating force within the muscle is caused by the interaction between actin and myosin filaments (Huxley, 1969). Briefly, when the motoneuronal drive to a muscle increases, new chemical bonds or "cross-bridges" are formed between the actin and myosin filaments. Consequently, an increase of muscle tension is generated concomitantly with an increase of muscle stiffness.

In spite of the complexity of the molecular processes underlying the generation of muscle force, at the macroscopic level a muscle can be characterized as a *tunable elastic element*. This behavior is illustrated in Figure 1 where tensions measured in the cat soleus muscle are plotted against muscle lengths at various levels of neural activations. The data were obtained by Rack and Westbury (1969) who devised a method for applying a set of stimuli to different portions of the same nerve. All the stimuli had the same frequency but different phases, in order to avoid synchronous stimulation of all muscle fibers. Then, for each tested frequency Rack and Westbury measured the steady-state tension corresponding to different muscle lengths.

Conceptually, muscle spring-like behavior can be summarized as follows:

1. For a fixed value of the motoneuronal input, the force generated at steady state is a function of muscle length.
2. This force/length function can be integrated to derive a potential function equivalent to the energy stored (or released) during an externally-imposed displacement.
3. Changes of neural input to the muscle smoothly affect the force/length relationship without drastically changing its shape.

Indicating by f , l and u the tension, the length and the neural input to a muscle and by E the stored

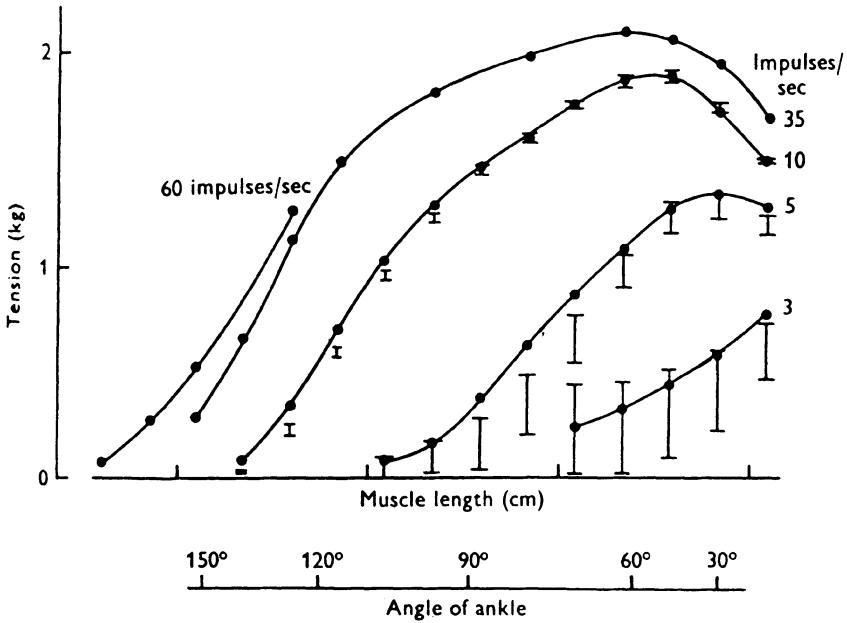


Figure 1:

Length-tension curves measured in the cat's soleus muscle. Different curves correspond to different stimulus rates. The vertical lines show the limits of tension fluctuations during synchronous stimulation. (From Rack and westbury, 1969).

elastic energy, the above statements are summarized by the expressions

$$f = f(l, u) \text{ and}$$

$$E = \int_{l_1}^{l_2} f(l, u) dl$$

(1)

We want to stress that in the above definition of spring-like behavior, the force/length relation is not required to be linear. As a matter of fact, the force/length curves in Figure 1 display considerable non-linearity. However, for any value of length, there is a locally linear relation between displacements, dl , and effort changes, df :

$$df = k(l) dl.$$

In the above expression, $k = \frac{\partial f}{\partial l}$, is the local muscle *stiffness* which, in general, depends upon muscle length. Whenever the stiffness is not zero, it is also possible to define a local *compliance*, $c = k^{-1}$, which maps an input force change into an output displacement.

Muscles are arranged about the joints in agonist-antagonist configurations. Then, a limb's posture is maintained when the forces exerted by the agonist and antagonist muscle groups are equal and opposite. When an external force is applied, the limb is displaced by an amount depending both upon the external force and upon the stiffness of the muscles¹. When the external force is removed, the limb should return to the original position. Experimental studies of motor behavior in humans and monkeys have confirmed that arm posture is indeed characterized by spring-like properties (Bizzi et al., 1984, Mussa-Ivaldi et al., 1985).

2.1. Virtual trajectories

This view of posture has been extended to the analysis of movement and trajectory formation (Bizzi et al., 1984, Hogan, 1984, McKeon et al., 1984, Flash, 1987.) The observation that posture is maintained by the equilibrium between the length-tension properties of opposing muscles led to the idea that movements result from a shift of the equilibrium point caused by a change in neural input. This hypothesis was first proposed by Feldman (1966).

Here, we discuss the implications of the moving equilibrium hypothesis in a computational perspective. To this end we start with a formal definition of the concept of virtual trajectory (a detailed discussion of related control issues can be found in Hogan, 1985). Let us first consider the case of a simple one-dimensional system characterized by the tunable-elastic behavior of Equation (1). Let the stiffness be defined, locally about a given length and at constant input, as $k = \frac{\partial f}{\partial l}$. The equilibrium position of the system, in the absence of external forces, is given by the condition $f(l, u) = 0$. A stable equilibrium position is characterized by negative stiffness (i.e. the forces about the equilibrium are attractive) whereas a positive stiffness implies an unstable equilibrium, similar to the equilibrium of a ball on the top of a hill.

If the input is a continuous function of time, $u(t)$, the static equilibrium condition becomes

$$f_e(l, u(t)) = 0 \quad (2)$$

where the subscript, e, indicates that f_e is the "elastic" or "steady-state" component of the total force acting on the system. If the environment does not prevent the system from moving as the input changes in time, the total force may also contain significant inertial and viscous components.

The above equation defines implicitly a (unique) map from $u(t)$ to l

$$l_e(t) = l(u(t)) \quad (3)$$

with $l_e(t)$ and $u(t)$ satisfying Equation (2), provided that the stiffness, k , at each equilibrium position,

¹For the present discussion it is immaterial that part of the muscle "restoring force" may be due to reflexes. In fact the static reflex response can be modelled by expressing the input u as the combination of a centrally originated signal, u_c , and a feedback signal, $u_r(l)$ which is function of l . Then, it is $f = f(l, u_r(l), u_c) = f(l, u_c)$ and the reflex behavior is equivalent to a stiffness component.

$l_0(t)$, is not zero. This is a direct consequence of a fundamental theorem on implicit functions².

The function $l_0(t)$ is the *virtual trajectory* of the system corresponding to the input function $u(t)$. A few important features of the virtual-trajectory concept are highlighted by this simple unidimensional example, in particular:

- it is not a "hypothesis" but a direct consequence of muscle spring-like properties.
- it can be regarded as a map from the control input (u) to an entity (l_0) which has the same physical dimension as a state component (l , the actual position of the system at any time).
- this map is given by an algebraic, rather than by a differential, equation³

It is interesting to observe that the fact that muscles are characterized by significant stiffness properties, i.e. that k is generally different from zero, implies that they cannot be considered as "good" force generators. An ideal force generator, as an ideal voltage source in an electric circuit, should have negligible output impedance. In other words, the output force should not depend upon the operating position but only upon the control input. This criterion has been followed in the design of many robotic actuators, such as the conventional torque motors. In contrast, biological evolution seems to have chosen a different route by developing muscles whose stiffness not only is significant but also depends on the level of neural activation. The notion of virtual trajectory transforms this feature into a strength rather than a weakness of the biological design. Actually, according to the implicit-function theorem, the actuators *must* have non-zero stiffness for mapping a sequence of control inputs into a sequence of well-defined equilibrium lengths. However, we will show that points of zero muscle stiffness do not necessarily constitute a problem in a multi-dimensional situation.

The extension of the virtual-trajectory concept to a multi-dimensional system is straightforward. Let us consider, for example, a serial arrangement of n limb segments interconnected by rotational joints. The configuration of this system is the array $\underline{q} = [q_1, \dots, q_n]^T$ of joint angles and the net torque is the array $\underline{T} = [T_1, \dots, T_n]^T$ of joint torques. Furthermore, the limbs are driven by a set of m muscles with $m > n$, i.e. there are more muscles than joints. Collectively, the muscles are represented by a length array, $\underline{l} = [l_1, \dots, l_m]^T$, by a force array $\underline{f} = [f_1, \dots, f_m]^T$ and by an array of input activations, $\underline{u} = [u_1, \dots, u_m]$. In the hypothesis that each muscle is characterized by tunable spring-like properties, the overall muscle behavior is described by the vector equivalent of Equation (1), namely

$$\underline{f} = \underline{f}(\underline{l}, \underline{u}) \quad (4)$$

We obtain the n dimensional joint-torque vector from the m -dimensional muscle-tension vector by

²For two variables, x and y , this theorem can be stated as follows. Let $F(x,y)$ be a function continuous with its first partial derivatives in a region R of the plane x,y and let the following conditions be satisfied: (a) $F(x^0, y^0) = 0$ at a point $P^0 = (x^0, y^0)$ in R and (b) the partial derivative $\frac{\partial F}{\partial y}$ is different from zero at the point P^0 . Then, the expression $F(x,y) = 0$ can be solved, i.e. a single-valued function can be defined, in the vicinity of P^0 .

³In general, for a dynamic system, a map of the form $\underline{x}' = g(\underline{x}, \underline{u})$ is defined. This map allows us to derive the future time course of the system state, $\underline{x}(t)$, i.e. the actual trajectory, given the input function $\underline{u}(t)$ and the initial value of the state, $\underline{x}_0$. Since the state equation is a differential equation its solution depends on the arbitrary selection of the initial conditions. In contrast, the virtual trajectory, being the result of an algebraic equation, does not contain arbitrary terms. Actually, the virtual trajectory defines the initial conditions for the state equations since the system is at static equilibrium before the onset of a movement.

multiplying the latter by the matrix of moment arms, $\underline{\mu}$. The elements of $\underline{\mu}$, μ_{ij} , are the partial derivatives $\frac{\partial l_i}{\partial q_j}$. The relation $\underline{l} = \underline{l}(\underline{q})$, which uniquely defines the muscle-length array as a function of the joint configuration is, in general, non linear. Therefore, the matrix of moment arms is also a function of joint configuration. Using Equation (4), the net joint torque vector becomes

$$\underline{\tau} = \underline{\mu}(\underline{q})^T \underline{f} = \underline{\mu}(\underline{q})^T \underline{f}(\underline{l}(\underline{q}), \underline{u}) = \underline{\tau}(\underline{q}, \underline{u}) \quad (5)$$

When the input array is a function of time, $\underline{u}(t)$, the static equilibrium condition $\underline{\tau} = \underline{0}$ defines implicitly the joint virtual trajectory as a function of $\underline{u}(t)$:

$$\underline{q}_e(t) = \underline{q}(\underline{u}(t)) \text{ such that } \underline{\tau}(\underline{q}_e(t), \underline{u}(t)) = \underline{0} \quad (6)$$

which is analogous to Equation (3.) From the fundamental theorem on implicit functions, the necessary and sufficient condition for the existence and uniqueness of the virtual trajectory $\underline{q}_e(t) = \underline{q}(\underline{u}(t))$ is that the determinant of the joint stiffness matrix $\underline{k}$, whose elements are $k_{ij} = \frac{\partial \tau_i}{\partial q_j}$ ($i, j = 1, \dots, n$), doesn't vanish⁴ at any equilibrium point.

This condition is equivalent to requiring non-zero stiffness in a one-dimensional systems. Here, a negative joint stiffness determinant corresponds to stable equilibrium whereas a positive determinant implies unstable equilibrium. Note, however, that asking for non-zero joint stiffness determinant is less stringent than requiring non-zero stiffness for each muscle. The stiffness about a joint is obtained from a weighted sum of the stiffness supplied by all the muscles operating about that joint. Hence, in order to have non-zero joint stiffness it is not necessary to require non-zero muscle stiffness. Actually, there is experimental evidence (Mussa-Ivaldi et al., 1985) that the multi-joint elastic forces during posture are attractive. In that case, the condition for defining virtual trajectories in hand or joint coordinates is satisfied.

The muscle spring-like properties allow mapping the input ($\underline{u}$) into a lower dimensional entity, the equilibrium arm configuration, which has the physical dimension of a position variable. Hence, the virtual trajectory, together with the joint stiffness, can be regarded as a *summary* of all the inputs involved in the execution of a movement. An important consequence of virtual trajectories is that they can be used to avoid the computation of inverse dynamics. The inertial and viscous components of the dynamics equations can be treated as "perturbing torques" which make the arm deviate from the desired path, specified as a sequence of static equilibrium configurations. At moderate speeds and accelerations, the difference between actual and virtual motion is small and can be neglected. Clearly, as movement speed and acceleration increase, limb inertia and viscosity are expected to cause larger deviations from the virtual trajectory. However, these deviations can be corrected by increasing the stiffness and/or by modifying the virtual trajectory itself.

The elastic properties of the actuators also provide stability to the planned movements (Hogan,

⁴The theorem of implicit functions is readily generalized to a system of n equations:

$$F_i(x_1, \dots, x_m, y_1, \dots, y_n) = 0 \quad ; \quad i = 1, \dots, n$$

by requiring that at a solution point $P^0 = (x_1^0, \dots, x_m^0, y_1^0, \dots, y_n^0)$ the functional (or Jacobian) determinant, $\frac{\partial F}{\partial y}$ is different from zero. Then, the above n equations can be solved for y_i , i.e. n functions $y_i = f_i(x_1, \dots, x_m)$ can be defined, in the vicinity of P^0

1985). In a computational approach based on the explicit solution of inverse dynamics, problems arise when external perturbations (e.g. hitting an obstacle) cause a deviation from the preprogrammed path since the dynamics must be recomputed or, alternatively, some special servo control mechanisms must be provided. In contrast, if movement is obtained by shifting the equilibrium position defined by elastic actuators, a deviation from the intended path simply results in larger restoring forces without need of computation or of particular control schemes (McKeon et al, 1984).

3. Passive motion

So far, we have how the actuator's elasticity can simplify the control tasks by providing an implicit map between a set of control input and a corresponding set of equilibrium configurations. These equilibrium configurations correspond to actual joint configurations when posture is maintained in the absence of external loads. During the execution of a movement, the static equilibrium configuration associated, at each time, with the control inputs is a moving "center of attraction" which generates a driving torque without any explicit dynamic computation. In the remainder of this paper we will show that the elastic properties of the actuators also lead to the solution of inverse kinematics problems in redundant motor systems.

The task of generating coordinated motion in a redundant mechanism has been conventionally described as an ill-posed inverse kinematics problem. Let us consider, for example, an extremely simple system composed of two linear elements connected in series, as shown in Figure 2A. The kinematic variables describing these elements are the lengths l_1 and l_2 . Then, the net system length, L , is derived from the element lengths through the "direct kinematics" equation:

$$L = l_1 + l_2 \quad (7)$$

Given the above expression, the inverse kinematics problem consists in finding the values of l_1 and l_2 corresponding to a desired total length, L . A geometrical interpretation of this problem is shown in Figure 2B, where the kinematic space of the system is represented as a two-dimensional coordinate frame. Each axis of this frame represents the length of one element. The oblique line is at a distance L from the origin and corresponds to the constraint equation (7). Then, all (and only) the pairs of values (l_1, l_2) corresponding to the points on this line are solutions of the inverse-kinematic problems for a desired displacement L . Optimization methods result in the selection of a specific point by adding some further constraint. For example, the point $D = (\frac{L}{2}, \frac{L}{2})$, at the intersection of the solution line with the perpendicular drawn from the origin, corresponds to a minimum-norm criterion.

The most common engineering approach to an underdetermined problem involves the definition of cost functions according to specific goals such as avoiding singular configurations (Bailleul et al., 1984), minimizing joint torques (Hollerbach and Suh, 1985), joint motion (Brooks, 1982) or kinetic energy (Kathib, 1983). Here, we present a different point of view, based on simulating passive motions within a field of elastic forces. This approach was introduced by Mussa-Ivaldi et al. (1988) and has been named the passive-motion paradigm. Essentially, the passive-motion paradigm

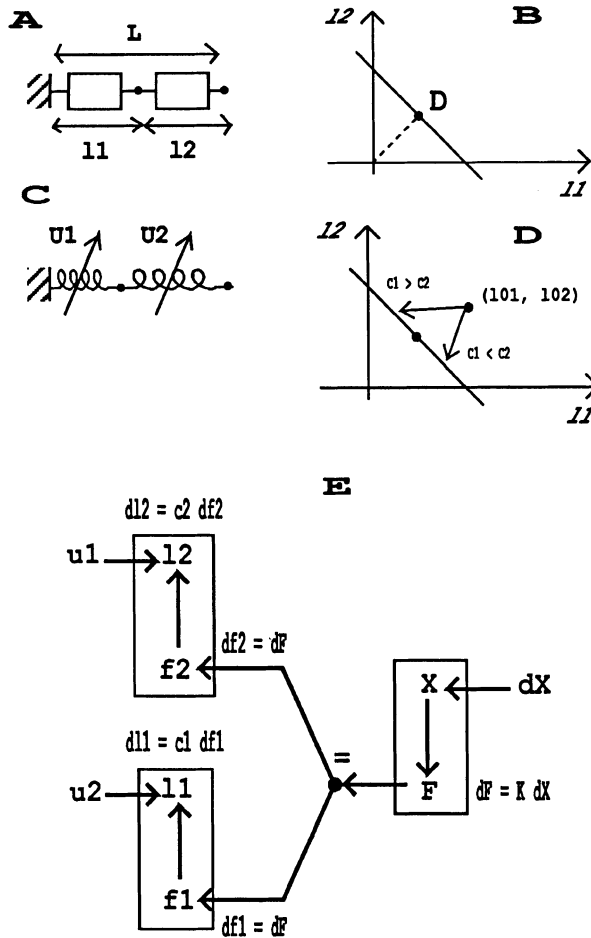


Figure 2: Passive motion in a simple system (see text.)

exploits the fact that a passive displacement of a physical system characterized by impedance properties always leads to a well-posed problem. This is the case regardless of the unbalance between the degrees of freedom of the system and the number of dimensions necessary to fully describe the imposed displacement. The regularization principle implicitly implemented by this physical solution is the minimization of potential energy. This fact is dramatically demonstrated by any elastic sheet, such as the surface of a drum which is characterized by an infinity of degrees of freedom. When a local displacement is applied at a point (e.g. by pushing with a pencil at the center of the drum,) the whole surface changes its configuration so that the strain energy is minimized.

In the example of Figure 2, the passive-motion paradigm requires the definition of the two components as elastic elements. Let's assume that these elements are tunable compliances (Figure 2C) with the following length/tension relations:

$$l_1 = c_1(u_1) f_1 + l_{01}(u_1) \quad (8)$$

$$l_2 = c_2(u_2) f_2 + l_{02}(u_2).$$

The parameters c_1 and c_2 are the element compliances. The parameters l_{01} and l_{02} are rest-lengths, that is the lengths assumed by each component when the tension is zero. Both compliance and rest-length depend upon each element's input in a known way. Let us also assume that the inputs are set to some initial value, u_1, INIT , u_2, INIT . Accordingly, the rest-length will assume the corresponding values $l_{01, \text{INIT}}$, $l_{02, \text{INIT}}$. If no external force is acting upon the system, it will settle to a "global" equilibrium position $l_{0, \text{INIT}} = l_{01, \text{INIT}} + l_{02, \text{INIT}}$. The net compliance at this location is $C = c_1 + c_2$ ($C \neq 0$.)

If at this point a total displacement dL is imposed on the system endpoint, a net elastic force is generated, $dF = C^{-1} dL$, opposite to the displacement. This is also the force "seen" at steady state by each of the elements, that is $df_1 = df_2 = dF$. Then, the two elements will be displaced by the amounts

$$dl_1 = c_1 dF = \frac{c_1}{c_1 + c_2} dL \quad (9)$$

$$dl_2 = c_2 dF = \frac{c_2}{c_1 + c_2} dL.$$

Thus, we have effectively an inverse-kinematic map from dL to the pair dl_1, dl_2 .

It is readily seen that this solution corresponds to the point of minimum potential energy compatibly with the kinematic constraint:

$$dl_1 + dl_2 = dL. \quad (10)$$

In fact, the potential energy, E , in the vicinity of the equilibrium configuration, $l_{01, \text{INIT}}$, $l_{02, \text{INIT}}$, is

$$E = \frac{dl_1^2}{2c_1} + \frac{dl_2^2}{2c_2} \quad (11)$$

Then, the first (second) Equation (9) is directly obtained by substituting (10) for dl_2 (dl_1) in (11) and finding the solution for $\frac{\partial E}{\partial l_1} = 0$ ($\frac{\partial E}{\partial l_2} = 0$.)

From this simple example it is possible to draw some important observations. First, the minimum norm solution, $dl_1 = dl_2 = \frac{dL}{2}$, is the "natural" response to a passive displacement when $c_1 = c_2$. By choosing different patterns of compliance one obtains different points on the solution set (Figure 2D) in a way which is strictly consistent with the notion of impedance: the more one element is compliant (relatively to the others), the more it will tend to contribute to the global motion. In general, different settings of the elastic properties associated with the degrees of freedom will yield different configurations of minimum potential energy, given the same externally imposed displacements. Since we are assuming that the control signals can *tune* the local elastic properties of the actuators, we have at our disposal a mechanism to generate different patterns of movements, given a single kinematic task.

The second observation concerns information processing. The computational scheme used for deriving the expressions (9) is inherently modular and distributed. In Figure 2E, each physical element is represented by a processing unit which computes its position and effort according to the environmental constraints and to its own elastic behavior. Furthermore, a computational unit is introduced representing the elastic behavior of the whole system: its effort and position variables are F and L respectively and its compliance is $C (= c_1 + c_2)$. Each element derives its own effort and position changes on the basis of locally available informations. When a displacement dL is imposed by the environment on the element representing the net system, the latter derives the effort change, dF , multiplying dL by its own stiffness. Then, each processing element sends this information to the elements representing smaller components. These compute their displacements independently of each other on the basis of their compliance. All the coupling information, in this case the net compliance, $C = c_1 + c_2$, is concentrated in the elastic element which represent the whole system.

It is possible to go one step further in the passive motion paradigm by letting each element change its input so that the new position is at equilibrium. Denoting by σ_i the sensitivity of element i 's length to a change in input, $\sigma_i = \frac{\partial l_i}{\partial u_i}$ and assuming that σ_i^{-1} exists, an input update $du_i = \sigma_i^{-1} dl_i$ applied independently to each controlled element is sufficient to bring the whole system at equilibrium in the new position $L' = L_{OINIT} + dL$.

As it happens with other types of network (Hopfield, 1982) the overall system behavior is characterized by a global potential function. In our case, this function is the elastic energy associated with the system's spring-like properties. Following the local input-update rule outlined above is equivalent to the global requirement that the transition between one set of inputs to the next is chosen so as to minimize the net change of potential energy, compatibly with the task that is, in this case, compatibly with the necessity of displacing the system equilibrium by an amount dL .

In the next section we will show that passive motion leads to well-defined inverse kinematic solution also in redundant kinematic systems characterized by non-linear geometries. Specifically, we will consider the problem of transforming a displacement of the hand dx , into the corresponding joint displacement dq .

4. Inverse kinematics

The kinematic transformations for a redundant serial chain, such as the human arm, are a set of non-linear equations that map the joint configuration $\underline{q} = [q_1, q_2, \dots, q_N]^T$ to a lower-dimensional set of task variables $\underline{x} = [x_1, x_2, \dots, x_M]^T$ ($M < N$.) We represent these equations as a single vector map

$$\underline{x} = \underline{x}(\underline{q}) \quad (12)$$

In the following discussion we will assume that this map is continuous and differentiable up to the second order within the entire workspace.

A typical choice for the task variables is the hand-position vector with respect to a base-frame. However, it should be stressed that the definition of $\underline{x}$ may change from one task to another. For example, we may carry an object without being concerned about its orientation. Then, $\underline{x}$ has only

three translational components. In contrast, we may be also specifying an orientation. In that case, $\underline{x}$ has three translational and three rotational components. Thus, the *degree of redundancy*, $N-M$, depends upon the nature of the task as well as upon the structure of the manipulator. An immediate consequence is that any general-purpose robot is bound to be used with some degree of redundancy whenever the task requirements become less stringent than those which guided the design process.

Given the kinematic transformation (12), the M -dimensional hand force $\underline{F}$ maps into the N -dimensional joint torque $\underline{T}$ as

$$\underline{T} = \underline{J}^T \underline{F} \quad (13)$$

where $\underline{J} = \frac{\partial \underline{x}}{\partial \underline{q}}$ is the Jacobian of the kinematics equations (12). The force transformation (13) is a direct consequence of the principle of virtual works. This principle states that the mechanical work (the inner product of force and displacement) is invariant by coordinate transformations. That is:

$$\underline{F}^T \underline{dx} = \underline{F}^T \underline{J} \underline{dq} = \underline{T}^T \underline{dq}.$$

Note that, while a displacement joint coordinates maps uniquely into a displacement in hand coordinates ($\underline{dx} = \underline{J} \underline{dq}$), the reverse is true for the forces: a tip force $\underline{F}$ maps uniquely into a torque vector $\underline{T}$. Thus, for a redundant serial mechanism we have two complementary ill-posed problems:

1. Finding the hand-force corresponding to an arbitrary set of joint torques. Since there are more constraint equations than unknowns, this problem may have no solutions. Joint torques cannot be selected arbitrarily.
2. Finding the joint configuration corresponding to a given hand position. There are more unknowns than equations and the problem has infinite solutions.

Here, we focus on the solution to the second problem which involves a transformation from task to joint coordinates. As is the case for the simpler spring system described in the previous section, a passive motion, $\underline{dx}$, imposed by the environment on the tip generates a unique configurational displacement $\underline{dq}$. This passive solution can be derived in two equivalent ways. In one case, one may search for the configuration that corresponds to a minimum of potential energy compatible with $\underline{dx}$. Alternatively, $\underline{dq}$ is obtained from the combination of three algebraic maps.

1. The first map is provided by the hand stiffness $\underline{K}_{hand}$ that transforms $\underline{dx}$ into a force change $\underline{dF}$ as:

$$\underline{dF} = \underline{K}_{hand} \underline{dx} \quad (14)$$

2. The second map is the transformation from $\underline{dF}$ to the corresponding torque-change, $\underline{dT}$. This map is obtained by differentiating the force/torque transformation (13):

$$\underline{dT} = \underline{J}^T \underline{dF} + \underline{dJ}^T \underline{F}. \quad (15)$$

The second term on the left-hand side is required to account for the fact that at steady-state, the configuration changes by an amount $\underline{dq}$. Since the Jacobian is configuration dependent, $\underline{J}$ changes by an amount $\underline{dJ}$. With simple calculations we obtain:

$$\underline{dJ}^T \underline{F} = \underline{\Gamma} \underline{dq}$$

where $\underline{\Gamma}$ is a $N \times N$ matrix which has the physical dimension of a joint stiffness and whose components are:

$$\Gamma_{ij} = \sum_{k=1}^M \frac{\partial^2 x_i}{\partial q_j \partial q_l} F_k. \quad (16)$$

Using this expression, the transformation (15) becomes

$$\underline{dT} = \underline{J}^T \underline{dF} + \underline{\Gamma} \underline{dq}. \quad (17)$$

3. Finally, the third map is the transformation from $\underline{dT}$ to $\underline{dq}$ which is provided by the joint stiffness, $\underline{c}$:

$$\underline{dq} = \underline{c} \underline{dT}. \quad (18)$$

Here, we assume that the compliance, $\underline{c}$, is the Jacobian of some arbitrarily defined control law (such as $\underline{q} = \underline{q}_0 + \underline{c}\underline{T}$.) This control law can either be explicitly implemented by some feedback mechanism or it can be a way for representing the steady-state behavior of some elastic actuator. In both cases we make the further assumption that $\det(\underline{c}) \neq 0$. Therefore, the causality of Equation (18) for deriving an output torque in response to an input displacement as:

$$\underline{dT} = \underline{k} \underline{dq} \quad (19)$$

where $\underline{k} = \underline{c}^{-1}$ is the joint stiffness.

We can combine the three differential maps (14), (17) and (18) as shown in Figure 3 to obtain a single transformation from $\underline{dx}$ to $\underline{dq}$:

$$\underline{dq} = (\underline{k} - \underline{\Gamma})^{-1} \underline{J}^T \underline{K}_{hand} \underline{dx}. \quad (20)$$

By imposing that this map is actually an inverse of the direct kinematics, $\underline{dx} = \underline{J} \underline{dq}$, we obtain the following expression for the hand stiffness:

$$\underline{K}_{hand} = (\underline{J} \underline{c}_{LIN} \underline{J}^T)^{-1} \quad (21)$$

where we have set $\underline{c}_{LIN} = (\underline{k} - \underline{\Gamma})^{-1}$. We call $\underline{c}_{LIN}$ the *linearized joint compliance*. Its inverse, the linearized joint stiffness, $\underline{k}_{LIN}$, differs from the actual joint stiffness by the correction matrix $\underline{\Gamma}$, which accounts for the position-dependency of the Jacobian matrix. Using this correction, the end-effector stiffness is obtained as a linear transformation of $\underline{k}_{LIN}$.

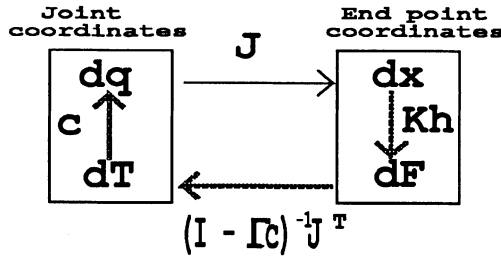


Figure 3:

Differential transformation graph for a redundant arm. The graph is fully connected: each node can be reached from any other node through a single path. The thicker arrows define a path from $\underline{dx}$ to $\underline{dq}$. This path is equivalent to a weighted generalized-inverse of the Jacobian, $\underline{J}$.

In summary, the elastic properties of the joint actuators expressed by Equation (18), provide a natural solution to the inverse kinematic problem for a redundant mechanism. Making use of $\underline{c}_{LIN}$ we can write this solution as the differential transformation

$$\underline{dq} = \underline{c}_{LIN} \underline{J}^T (\underline{J} \underline{c}_{LIN} \underline{J}^T)^{-1} \underline{dx} \quad (22)$$

which is a weighted generalized-inverse of $\underline{J}$. This expression characterizes the change of configuration corresponding to a quasi-static displacement of the endpoint. Therefore it is equivalent to minimizing potential energy while satisfying the constraint imposed by the motion of the end-point.

5. Passive motion is integrable

We have shown that passive motion provides a local solution to the inverse kinematic problem (Equation (22)) as a weighted generalized-inverse of the Jacobian matrix. The general expression for the weighted generalized-inverse of $\underline{J}$ is

$$\underline{P}_w = \underline{w} \underline{J}^T (\underline{J} \underline{w} \underline{J}^T)^{-1} \quad (23)$$

where $\underline{w}$ is some $N \times N$ weight matrix. With this generalized inverse, the configuration change $\underline{dq} = \underline{P}_w \underline{dx}$ is guaranteed to satisfy the kinematic constraint

$$\underline{dx} = \underline{J} \underline{dq}$$

while, at the same time, minimizing the quadratic form

$$E_w = \underline{dq}^T \underline{w}^{-1} \underline{dq}.$$

In particular, when $\underline{w}$ is the unit matrix, $\underline{I}$, the quadratic form E_I is the norm of the configuration change, $\underline{dq}^T \underline{dq}$. Thus, the generalized inverse $\underline{P}_I = \underline{J}^T (\underline{J} \underline{J}^T)^{-1}$ which is also known as the Moore-Penrose pseudoinverse of $\underline{J}$, corresponds to a minimum-norm criterion (Ben Israel and Greville, 1980). The map corresponding to an externally imposed displacement is equivalent to the application of a generalized pseudoinverse $\underline{P}_{c_{LIN}}$. This is consistent with the physical requirement that the passive configuration change is constrained by a minimum potential energy criterion.

There is no a priori guarantee that a weighted generalized inverse of $\underline{J}$ is *integrable*, i.e. that within some domain of the workspace, $\underline{P}_w$ is the Jacobian of some global inverse-kinematic function mapping any given position $\underline{x}$ into a single configuration $\underline{q}$. Whenever such a map exists, then iterating its Jacobian from a starting position $\underline{x}_0$ to a final position $\underline{x}_1$ results into a final configuration $\underline{q}_1$. The latter depends upon the initial configuration $\underline{q}_0$ *but not upon the particular workspace path chosen to join $\underline{x}_0$ to $\underline{x}_1$* . Equivalently, a necessary and sufficient condition for a local generalized-inverse map to be integrable is that the iteration of this map along any closed path brings the system back to the same starting configuration. A number of investigations (Liegeois, 1977, Klein and Huang, 1983, Hollerbach and Suh, 1985, Wampler, 1987) have demonstrated that the most common generalized-inverse operators *do not* provide integrable maps. Most notably, Klein and Huang (1983) showed that iterating the Moore-Penrose pseudoinverse along closed paths can produce a variety of non-cyclic behaviors. In some cases, as the hand passes repeatedly through the same point, the joint configuration changes until a limit configuration is reached. However, this limit configuration may depend upon the direction of travel around the path. In other cases, the configuration keeps changing without reaching any limit.

Mussa-Ivaldi and Hogan (1989) have recently demonstrated that the weighted generalized inverse, $\underline{P}_{c_{LIN}}$, corresponding to an externally imposed displacement is integrable within any simply connected domain in which the manipulator end-point doesn't loose mobility (i.e. in which

$\det(\underline{J} \underline{C}_{LIN} \underline{J}^T) \neq 0$.) This result is consistent with the physical notion that the minimum potential energy constraint is sufficient to uniquely define a configuration corresponding to a given location of the end-point. Consider, for example a redundant serial kinematic chain. Let each joint be characterized by a fixed torque/angle relation so that the linkage is at equilibrium at a configuration $\underline{q}_0$ corresponding to a tip-location $\underline{x}_0$. This relation can be simply obtained by placing a "winding spring" on each joint. It is intuitively evident that if we force the tip along a closed path terminating back to $\underline{x}_0$, the whole mechanism will return to the starting configuration. It is also intuitive that the only way to achieve a different configuration is by "winding" the joint springs. This can be achieved by moving the tip along a path which encloses some singular points of the workspace, that is carrying out the iteration in a non-simply-connected domain.

From an algebraic point of view, the key to the integrability of passive motion is provided by the matrix $\underline{\Gamma}$ whose elements are given by Equation (16). This matrix is an impedance component which contributes to the expression of $\underline{C}_{LIN}$. Let us consider a situation in which the following control law is applied to each joint:

$$T_i = k (q_i - q_{0i})$$

(k is the same scalar constant for all the joints.) Then, the net joint stiffness is proportional to the unit matrix. Let us assume that the mechanism is at the initial equilibrium configuration $\underline{q}_0$, corresponding to the initial end-point location $\underline{x}_0$. As the end point is forced by the environment to follow a closed path, the configuration undergoes a sequence of displacements

$$d\underline{q} = (\underline{I} - \underline{\Gamma}(\underline{q})) \underline{J}^T (\underline{J} (\underline{I} - \underline{\Gamma}(\underline{q}))^{-1} \underline{J}^T)^{-1} d\underline{x}. \quad (24)$$

Initially, $\underline{\Gamma}$ is equal to zero since the end point is at equilibrium ($F=0$). Thus, the initial displacement is simply given by the Moore-Penrose pseudoinverse, $\underline{P}_1$. As the tip is moved away from equilibrium, the elastic force increases and, consequently, the $\underline{\Gamma}$ term also increases. On the way back, $\underline{\Gamma}$ decreases. When the end-point has returned to the initial position $\underline{x}_0$, the whole mechanism is guaranteed to be at the starting configuration $\underline{q}_0$ (provided that the path did not enclose any singularity). In contrast, if the displacements around the whole path were calculated using the Moore-Penrose pseudoinverse (i.e. with $\underline{\Gamma} = 0$ everywhere) then, returning to $\underline{x}_0$, the configuration would have been different from the starting value $\underline{q}_0$. Thus, the matrix $\underline{\Gamma}(\underline{q})$ is a configuration-dependent correction that is equivalent to a "memory" of the initial conditions.

6. Joint-limits and singularities

In order to illustrate the computational value of integrability, let us consider the problem of joint-limit avoidance. All the biological joints are characterized by a limited range of motion. One possible role of joint limits is to help preventing the arm from falling into kinematic singularities located in the interior of the workspace. A kinematic singularity is a configuration in which $\text{rank}(\underline{J})$ is less than the task-space dimension. In this case the manipulator loses mobility by becoming "infinitely stiff" in one or more directions. For example with a three-link planar arm, singular configurations are reached when the elbow angle is 0 or 180 degrees (forearm parallel to the upper arm) and the wrist angle is 180 degrees (hand folded back onto the forearm). Since the physiological elbow and wrist cannot reach these angles, the corresponding singularities are avoided. However, joint limits

introduce another type of singularity: the end-point stiffness becomes singular whenever the joint compliance matrix loses rank (more precisely, whenever $\text{rank}(\underline{C}_{LIN}) < M$). Since the joint limits are angles at which the joints reach zero compliance, their presence introduces new singular points.

In the context of passive motion, one way to avoid joint limits is to design the stiffness of each joint so that it increases as the joint approaches a limit angle. Thus in a passive displacement of the end point, when a joint gets near to a limit a larger portion of the motion is taken over by those other joints that are closer to their midrange. Following this rationale, one would expect that singular points are reached only when several joints are simultaneously close to their limits. This approach is illustrated in Figure 4 for a three-link planar manipulator. The shaded area near each joint indicates the range of motion (Figure 4A). Within this range the stiffness is set as (Figure 4B):

$$k(q) = \cos^2(A(q - q_{MID})) \quad (25)$$

$$A = \pi/(q_{MAX} - q_{MIN}) ; q_{MID} = (q_{MIN} + q_{MAX})/2.$$

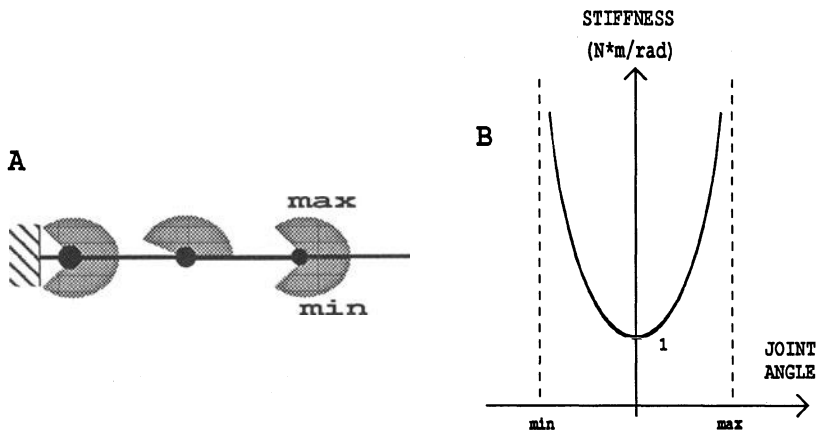


Figure 4:

Three-joint planar manipulator with limited joint ranges. A) The shaded areas indicate the range of motion of each joint. B) Joint limits are implemented by the joint-stiffness increasing at the extremes of the ranges of motion.

Figure 5 shows a comparison between two passive motions from the same starting configuration to the same end-point locations. In Figure 5A, the manipulator joints have equal and constant stiffness through the workspace. At the end of the movement the joint configuration ($q_1 = 39^\circ$, $q_2 = 168^\circ$, $q_3 = 27^\circ$) is close to a kinematic singularity ($q_2 = 180^\circ$, $q_3 = 0^\circ$). In Figure 5B, the same manipulator has variable joint stiffness, according to Equation (25). Thus, the motion of the elbow has been significantly reduced at the expenses of larger displacements of the wrist and of the shoulder. Consequently, the final configuration ($q_1 = 66^\circ$, $q_2 = 150^\circ$, $q_3 = 36^\circ$) is at a larger distance from the singularity.

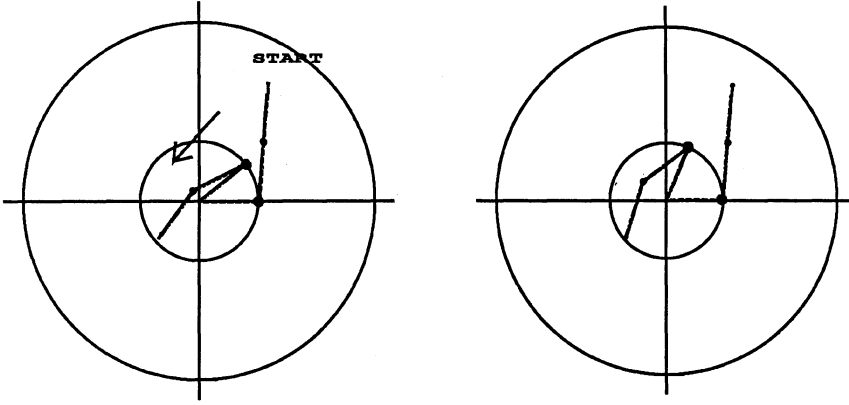


Figure 5:

Dangerous locations. A) The arm moves from the start to the end locations with uniform joint stiffness. B) The same endpoint movement (from the same starting configuration) is performed with the joint stiffness increasing towards the joint limits. (Joint ranges. Shoulder and wrist : $\pm 140^\circ$. Elbow: $0^\circ - 170^\circ$.)

We simulated the passive movements by iterating an integrable generalized-inverse of the Jacobian matrix (Equation (22)). For joint-limit avoidance, the expression of this generalized inverse is:

$$\underline{P}_{c_LIN} = (\underline{k}(q) - \underline{\Gamma})^{-1} \underline{J}^T (\underline{J} (\underline{k}(q) - \underline{\Gamma})^{-1} \underline{J}^T)^{-1} \quad (26)$$

where $\underline{k}(q)$ is a diagonal joint-stiffness matrix whose elements are given by Equation (25). As it has been shown by Mussa-Ivaldi and Hogan (1989) the integrability of $\underline{P}_{c_LIN}$ is entirely a consequence of the $\underline{\Gamma}$ matrix. This point is illustrated by the following example. Let us consider the generalized inverse, $\underline{P}_c$, that is obtained from $\underline{P}_{c_LIN}$ by removing the $\underline{\Gamma}$ matrix :

$$\underline{P}_c = \underline{k}(q)^{-1} \underline{J}^T (\underline{J} \underline{k}(q)^{-1} \underline{J}^T)^{-1}. \quad (27)$$

$\underline{P}_c$ is a weighted generalized-inverse, whose weight matrix depends upon the joint configuration according to Equation (25). Then, consider the task of moving the end point of the three-link manipulator on a closed target pattern, as it is shown in Figure 6. The starting target, 1, corresponds to the initial configuration ($q_1 = 0^\circ$, $q_2 = 85^\circ$, $q_3 = 0^\circ$). From this configuration we iterated $\underline{P}_c$ until the hand reached target 2. Then we continued the iteration of $\underline{P}_c$ on a straight line to target 3, and finally, the path terminated at target 1. Note that the final and the initial configurations are different, in spite of the absence of singularities within the triangle 1-2-3. Thus, $\underline{P}_c$ is not integrable.

This result is explained as follows. As the tip goes to target 2, the elbow joint gets close to its lower limit (0°). Target 2 is indeed near the boundary of the workspace, where the inverse

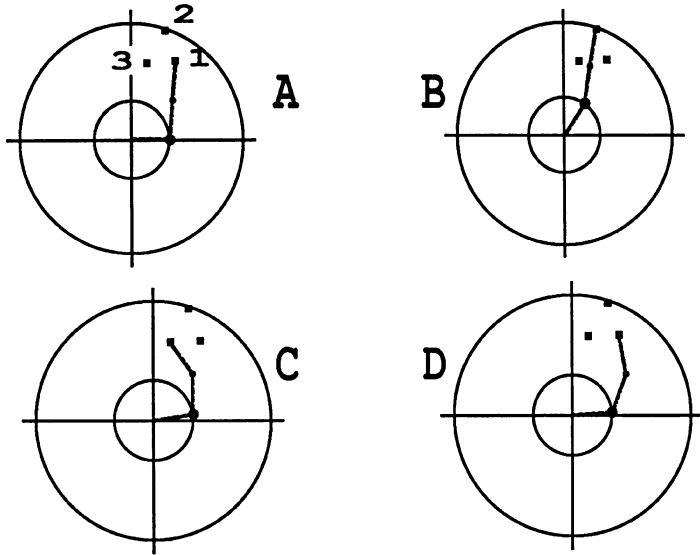


Figure 6:

Joint-limit avoidance. Iteration of a non-integrable generalized inverse around a closed path. The arm does not return to the starting configuration (see text.)

kinematics is unique and singular. At this location the elbow has a large stiffness, compared to the other joints. Thus, when the hand moves to target 3 and then back to target 1, the elbow angle tends to remain constant. Consequently the final position is reached with an elbow angle that is smaller than the initial value at the same tip location. Using $\underline{P}_c$, once a joint gets near to its limits it tends to remain "stuck", since $\underline{P}_c$ has no memory.

In contrast, when we used $\underline{P}_{c_{LIN}}$ to generate the same motion of the tip, the joint configuration returned to the initial value as the tip moved back to target 3 (Figure 7). Note that the configuration obtained at target 2 by iterating $\underline{P}_{c_{LIN}}$ is nearly identical to that obtained by iterating $\underline{P}_c$. This result is merely a consequence of the fact that target 2 is close to the workspace boundary (where the manipulator "loses" its redundancy.) Therefore, also with $\underline{P}_{c_{LIN}}$ the elbow stiffness at target 2 is larger than the other joint stiffness components. However, this differential operator now contains a memory of the initial conditions, the matrix $\underline{\Gamma}$, which is sufficient to obtain a larger elbow motion when the tip returns to more internal regions of the workspace.

7. Passive behavior and active control

So far we have represented the kinematic tasks (that is the desired motions of the endpoint) as simulated passive-motions that are imposed by the environment upon the end-effector. The manipulator was assumed to be characterized by fixed elastic properties (Equation (18) or (25)).

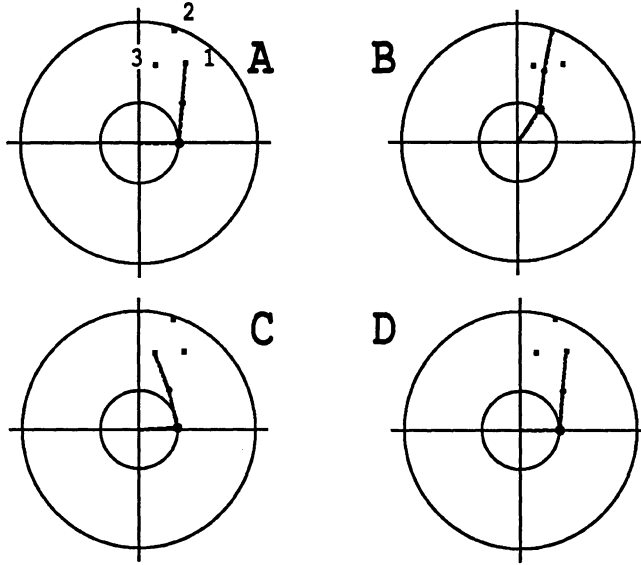


Figure 7:

Joint-limit avoidance. Iteration of an integrable generalized inverse around the same path of Figure 6. The arm returns to its starting configuration (see text.)

Then, the degrees of freedom move in such a way as to minimize the potential energy stored in the compliance. We have shown that the matrix $\underline{\Gamma}$ is essential to simulate this process correctly by taking into account the effects of the non-linear geometries which become significant as the endpoint moves away from the equilibrium position. Thus, we have shown that $\underline{\Gamma}$ is equivalent to a memory term.

An alternative point of view is to consider $\underline{\Gamma}$ as an impedance instead of as a geometrical term ($\underline{\Gamma}$ has indeed the physical dimension of a joint stiffness.) Let us assume that the desired motion is expressed as a sequence of equilibrium positions. To achieve this the N configuration variables, q_i , must depend not only upon the generalized forces, but also upon N control inputs, u_i . The required behavior of the actuators is then characterized by a controllable compliance function of the form:

$$\underline{q} = \phi(\underline{\Gamma}, \underline{u}). \quad (28)$$

The equilibrium configuration associated, at steady state, with an input vector $\underline{u}$ is

$$\underline{q}_0(\underline{u}) = \phi(\underline{0}, \underline{u}). \quad (29)$$

A change, $d\underline{u}$, of the input causes the equilibrium configuration to be updated by an amount:

$$d\underline{q}_0 = \underline{\sigma} d\underline{u} \text{ with } \underline{\sigma} = \frac{\partial \phi}{\partial \underline{u}}. \quad (30)$$

Here, $\underline{\sigma}$ is a local sensitivity matrix which we will assume to be non-singular ($\det(\underline{\sigma}) \neq 0$). Then, as the input changes smoothly in time, the above equation defines a sequence of static equilibria - a virtual trajectory.

The inverse kinematic problem becomes that of finding an appropriate sequence of inputs $\underline{u}(t)$, given a desired trajectory of the endpoint, $\underline{x}_0(t)$. One way to do this is to simulate externally-imposed displacements which will drive the joints away from equilibrium; then, at the end of each displacement, the input is modified to set equilibrium at the new manipulator configuration. This method which has been called the passive-motion paradigm leads to a distributed representation of motor redundancy (Mussa-Ivaldi et al, 1988). Starting from an equilibrium position, $\underline{x}_0$ (corresponding to $\underline{g}_0(\underline{u})$), using the generalized-inverse $\underline{P}_{c_LIN}$ ($\underline{c} = \frac{\partial \phi}{\partial \underline{T}}$) corresponds to iterating a change of equilibrium, $d\underline{g}_0 = \underline{P}_{c_LIN} d\underline{x}_0$, by updating the input with $d\underline{u} = \underline{\sigma}^{-1} d\underline{g}_0$. However, since this process occurs about equilibrium, the matrix $\underline{\Gamma}$ is zeroed at each step, and $\underline{P}_{c_LIN} = \underline{P}_c$. Therefore there is no guarantee that the input update is integrable, unless $\underline{c} = \underline{c}'_{LIN}$ for some compliance matrix $\underline{c}'$. This condition corresponds to requiring that at each equilibrium configuration, the joint compliance, $\underline{c}$, is:

$$\underline{c} = (\underline{c}'^{-1} - \underline{\Gamma})^{-1} \quad (31)$$

where $\underline{c}'$ is the compliance associated to some known differentiable function $\phi'(\underline{T})$.

8. Summary and conclusions

In this paper we have considered the computational value of the mechanical properties which characterize the biological actuators. In particular, we have focussed on the force/length relation which characterizes the behavior at steady state of human muscles. The fact that muscle force has a significant dependence upon muscle length constitutes a major departure of muscle's behavior from the canonical idea of a "good" robotic actuator: a force generator with negligible impedance. However, here we have shown that instead of implying a computational burden, the muscle properties offer simple solutions to some major problems such as (a) the problem of mapping a large set of control inputs into the equilibrium posture of a limb, (b) the problem of generating movements without computing the inverse dynamics and (c) the inverse kinematic problem for redundant systems.

Muscle elasticity provides a unified approach to posture and movement. Postures are equilibrium configurations resulting from the balance of agonist and antagonist muscles. These configurations act as points of attraction whenever an external perturbation generates some displacement. At the same time, the equilibrium posture is associated to a pattern of neural commands directed to each muscle. Thus, as the neural inputs to the muscles change in time, the static equilibrium shifts along a "virtual trajectory" and acts as a center of attraction interacting with limb inertia and viscosity. Here, we have shown that the virtual trajectory provides a unique map from a high-dimensional set of neural-activation variables to lower-dimensional kinematic variables: the joint equilibrium configuration and the equilibrium position of the endpoint. The condition for this mapping to be defined is simply that the net joint and hand stiffnesses have non-zero determinants. So far, this condition has been found to be satisfied in multi-joint posture (Mussa-Ivaldi et al., 1985).

Elastic properties also provide a natural solution to ill-posed coordination problems arising in redundant systems - systems with an excess of degrees of freedom with respect to the number of kinematic variables which fully describe a task. It is widely acknowledged that adding degrees of freedom to a manipulator can significantly improve its dexterity, that is its ability to perform different tasks in a wide range of environmental conditions. However, redundancy also introduces new challenging problems. One is that of computing the different joint-coordination patterns corresponding to a single end-effector task. Another is to ensure that local inverse-kinematic solutions correspond to well-defined global maps. Both problems are solved by the physics of elastic actuators when a passive motion is imposed by the environment to the endpoint.

Passive motion offers a natural solution to ill-posed kinematic problems. This solution rests upon the principle that the potential energy is minimized compatibly with the kinematic constraints. Furthermore, the representation of passive motion as an incremental process can be integrated to provide global inverse-kinematic functions (Mussa-Ivaldi and Hogan, 1989). Therefore these solutions do not depend upon the path that is followed to reach an endpoint location from a given starting point. A significant aspect of passive motion is that its integrability can be entirely attributed to a specific operator, the gamma matrix. This operator is equivalent to a joint-stiffness correction that can be applied to a virtual-trajectory controller. Thus it is possible to generate active motion which mimics the integrable passive behavior.

ACKNOWLEDGMENTS: This work was supported by the National Institute of Neurological Disease and Stroke Research, Grant NS09343 and by the Office of Naval Research, Grant N00014/88/K/0372.

References

- Baillieul, J, Hollerbach, J, M and Brockett, R, W (1984) *Programming and Control of Kinematically Redundant Manipulators*. Proc. of the 23rd IEEE Conf on Decision and Control, Las Vegas, Nevada, Dec 12-14: 768-774
- Ben-Israel, A e Greville, T, N (1980) *Generalized Inverses: Theory and Applications*. R. E. Krieger Publishing Co., New York
- Bizzi, E, Polit, A and Morasso, P (1976) *Mechanisms Underlying Achievement of Final Head Position*. J. Neurophysiol. 39:435-444
- Bizzi, E, Accornero, N, Chapple W and Hogan, N (1984) *Posture Control and Trajectory Formation During Arm Movement*. J. of Neurosci. 4:2738-2744
- Brooks, T, L (1982) *Optimal Path Generation for Cooperating or Redundant Manipulators*. Proc 2nd Int. Computing and Engineering Conf., San Diego, CA: 119-122
- Feldman, A, G (1966) *Functional Tuning of Nervous System with Control of Movement or Maintenance of a Steady Posture. II. Controllable Parameters of the Muscles. III. Mechanographic Analysis of the Execution by Man of the Simplest Motor Task*. Biophysics 11:565-578;766-775
- Flash, T (1987) *The Control of Hand Equilibrium Trajectories in Multi-joint Arm Movements*. Biol. Cybern. 57:257-274

- Gordon, A, M, Huxley, A e Julian, F, J (1966) *The Variation in Isometric Tension wit Sarcomere Length in Vertebrate Muscle Fibers*. J. Physiol. 184:170-192
- Hogan, N (1984) *An Organising Principle for a Class of Voluntary Movements*. J. Neurosci. 4:2745-2754
- Hogan, N (1985) *The Mechanics of Multi-joint Posture and Movement Control*. Biological Cybernetics 53:1-17
- Hollerbach, J, M and Suh, K, C (1985) *Redundancy Resolution of Manipulators Through Torque Optimization*. Proc. IEEE Int. Conf. on Robotics and Automation, St. Louis, MO, pp.1016-1021
- Hopfield, J, J (1982) *Neural Networks and Physical Systems With Emergent Collective Computational Abilities*. Proc. Natl. Acad. of Sciences, USA, 79:2554-2558
- Huxley, H, E (1969) *The Mechanism of Muscular Contraction*. Science 164:1356-1366
- Kathib, O, *Dynamic Control of Manipulators in Operational Space*. 6th IFToMM Congress on Theory of Machines and Mechanisms, New Dehli, December 15-20
- Klein, C, A and Huang, C, H (1983) *Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators*. IEEE, Trans. System Man and Cybern., SMC-13, pp. 245-250.
- Liegeois, A (1977) *Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms*. IEEE Trans. System Man and Cybernetics, SMC-7: 868-871
- McKeon, B, Hogan, N and Bizzi E (1984) *Effect of Temporary Path Constraint During Planar Arm Movements*. Abstr. 14th Ann. Conf. Soc. for Neurosci. (Anaheim CA Oct. 10-15): 337
- Mussa-Ivaldi, F, A, Hogan, N and Bizzi, E (1985) *Neural, Mechanical and Geometric Factors Subservient Arm Posture in Humans*. J. of Neurosci. 10:2732-2743
- Mussa-Ivaldi, F,A, Morasso, P and Zaccaria, R (1988) *Kinematic Networks. A Distributed Model for Representing and Regularizing Motor Redundancy*. Biol. Cybern. 60:1-16
- Mussa-Ivaldi, F, A and Hogan, N (1989) *Solving Kinematic Redundancy with Impedance Control: a Class of Integrable Pseudoinverses*. Proc. IEEE Intl. Conf. on Robotics and Automation, Scotsdale, AR: 283-288
- Nichols, T, R and Houk, J, C (1976) *Improvement in Linearity and Regulation of Stiffness That Results from Actions of Stretch Reflex*. J. Neurophysiol. 39:119-142
- Rack, P, M, H and Westbury, D, R (1969) *The Effects of Length and Stimulus Rate on Tension in the Isometric Cat Soleus Muscle*. J. Physiol. 204:443-460
- Wampler, C, W (1987) *Inverse Kinematic Functions for Redundant Manipulators*. Proc. IEEE Intl. Conf. on Robotics and Automation, Raleigh, NC:610-617

Motion Control in Intelligent Machines

A. Meystel

Department of Electrical and Computer Engineering
Drexel University, Philadelphia, PA 19104

Abstract

Motion control in Intelligent Machines is a part of their system of Intelligence. The system of Intelligence is viewed as a nested heterarchical structure with control loops associated with its nested mechanism of decision making. NICS philosophy is outlined, and an attempt is made to consider an operator of information representation and refinement serving the decision making processes. Different task-decomposition routines are considered supporting decision making within the level of resolution and decision-making processes which determine the interaction of adjacent resolution levels. NICS* - operator is introduced applicable to systems of Intelligent Control.

1. Introduction

1.1 Motivation

Motion control in intelligent machines is a part of their overall functioning. Until now we do not have a uniform well established approach to dealing with functioning of intelligent machines. Nested Intelligent Control Structure (NICS) proposed in this paper is inspired by and is a further development of the nested approach to decision making in NASREM. The latter has been introduced first in [1] and then its developments were presented in [2-4]. It turned out that NASREM is applicable for a wide variety of systems in Robotics and Computer Integrated Manufacturing.

The core of the NASREM concept can be formulated as follows: any system can be modeled as a sequence of three subsystems¹: Sensory Processing (Perception), World Model (Knowledge Organization), and Task Decomposition (Planning/Control)². Each of these subsystems is a hierarchy, and all three hierarchies communicate horizontally at different levels of resolution. The **Goal** of functioning is coming at the top and is subjected to subsequent decomposition into a hierarchy of tasks, and **Action** is presumably emerging at the bottom of the subsystem of Task Decomposition. External information is presumably arriving at the bottom of

¹ Structurally, these subsystems are built as **heterarchies**, i.e. hierarchies with links among the nodes at each level of the structure. We will never talk about hierarchies, only about heterarchies in order to underline this property of a heterarchy: having an interrelated structure horizontally at each particular level.

² Planning/control system is supposed to produce the heterarchy of task decomposition. A string of tasks at each level of the hierarchy is what we call "the control law" for this level. So, in an intelligent machine the task decomposition is never given to the controller: it should be generated by the controller.

the pyramid of sensory processing. All subsystems are somewhat connected to the Global Memory.

The information processing hierarchies are built according to general views presented in [5]. Interaction among the three subsystems (Perception, Knowledge Representation, and Planning/Control) is similar to processes characteristic for structures described in [6]. However, the original NASREM architecture [1-4] does not elaborate on the Planning/Control processes alluding to them as to processes of Task Decomposition which is understood as a set of solely inter-level activities. On the other hand, although NASREM presumes dealing with a real closed-loop systems, it is considered only for a high-resolution level. In general, the control part of NASREM is not described in detail; for example, it never makes a difference between feed-forward and feedback parts of the systems functioning. Finally, it is usually overlooked that NASREM methodology can (and should) be recursively applied to each component of its own structure.

1.2 Organization of paper

This paper focuses on the following five issues: 1) a concept of *intelligent machine* based upon the idea of *intelligence*, 2) general foundations of knowledge representation in intelligent machines, 3) peculiarities of the hierarchy of task decomposition, 4) synthesis of task-strings, or controls (planning/control processes), and 5) the closed-loop character of operation. This allows for building a model of Nested Closed Loop Control Structure based on NASREM-like subsystems of Knowledge Representation, Knowledge Acquisition, and Decision Making. We will call it **NICS: Nested Intelligent Control Structure**. NICS structure which can be considered explanatory for cognitive processes, and is instrumental for design of Intelligent Control Systems. Section II focuses on commonality of metaphors utilized for modeling similar processes in machines and living organisms. Techniques of dealing with unstructured and unclear knowledge are addressed in Section III. Different types of task decomposition and synthesis of decisions are considered in Section IV. Generation of feedforward controls and shaping external feedback loops (real and conceptual) are discussed in Section V. Finally, in Section VI the future problems are discussed for systems based upon a multiplicity of NICSs.

2. Intelligent System

2.1. Intelligence

Intelligent System is defined here as a system with (which is *governed* by) Intelligent Controller. *System* is understood as the collection of *objects* with *relationships* among these objects. This definition is a recursive one and can be applied to any object within the system which can be considered a system by itself. Thus a mechanism of *generalization* is implied which allows for neglecting inner mechanics of the system if it is considered an object. In other words, a multiresolutional technique of analysis is applied which entails multiresolutional knowledge acquisition and representation, multiresolutional decision making, etc.

Intelligent System is postulated to be governed as to achieve the goal. Goal can be assigned externally, or generated within the system. The device for governing the Intelligent System will be

called an Intelligent Controller. Control is understood as a subsystem, (its activities, and/or its output) which serves for generating plans, tasks, subtasks, commands (actually, *control* plans, *control* tasks, *control* subtasks, *control* commands) which are required to provide for a proper functioning of the system. This subsystem can be distributed over the whole *intelligent machine* under consideration. Intelligent Control is defined as Control which is generated by a controller with a property of Intelligence. Definitions of Intelligence can be found in [59].

Robotic Intelligence (or Machine Intelligence³) can be considered a property of the robot's computer controller, or a computer controller of an Intelligent Machine **to perform assignments in an uncertain environment** (level 1) **with no human involvement**, and **to independently develop new assignment for subsequent activities** (level 2). The term "uncertain" should be understood in a sense: not accounted for, not predicted by the assignment (level 1), or by the stored knowledge as well as by an assignment (level 2). Clearly, both of the levels of intelligence are associated with certain degrees of perception, knowledge organization, and decision making (see Figure 1). The structure is divided in two devices: intelligent controller computer and system to be controlled. Later we will see that this division is equivalent to dividing the whole world into two closely related Worlds: **WORLD OF REFLECTION** existing only in IC (brain's) imagination, and **WORLD OF REALITY** which exists as a hardware set. (It is interesting that these two worlds are mirror reflection of each other).

The following definition of intelligence will be accepted for the subsequent presentation: *intelligence is the ability to cope successfully with specified as well as nonspecified problems*. In the area of NC-machines most of the circumstances are presumed to be specified. In the area of robotics when the goal of operation is assigned something is always not specified. If the Problem is scrupulously specified, then the environment might be not completely specified. If both, problem and environment are specified, it always so happen that part of the specifications is at a level of generality which is leaving many details not specified, not accounted for.

2.2 Issues of Interest

Assignment is understood as a set of outputs of the goal generator (external and/or internal). External environment is a part of the "world" which can be controlled by the Intelligent Controller (IC) only via "actions" and does not received commands from IC directly. On the other hand, these actions can be considered "commands", and the external environment is becoming a part of the control model. Actions imply change which generates a need in introducing an idea of "state" and probably, a "state space" for describing the system and its environment as a string of the "snapshots". Then the system is being controlled by IC with a goal as its assignment and inputs for the system that the IC is destined to generate. Then this is a system for actuating the control profiles generated by the IC and affecting the environment by "actions" produced by its actuators. This individual intelligent machine contains subsystems which in turn are divided in parts down to the primitive which cannot be divided anymore. On the other hand a particular intelligent machine

³ Discussion of differences between human and machine intelligence is out of the scope of this paper. Briefly this issue is addressed in [59].

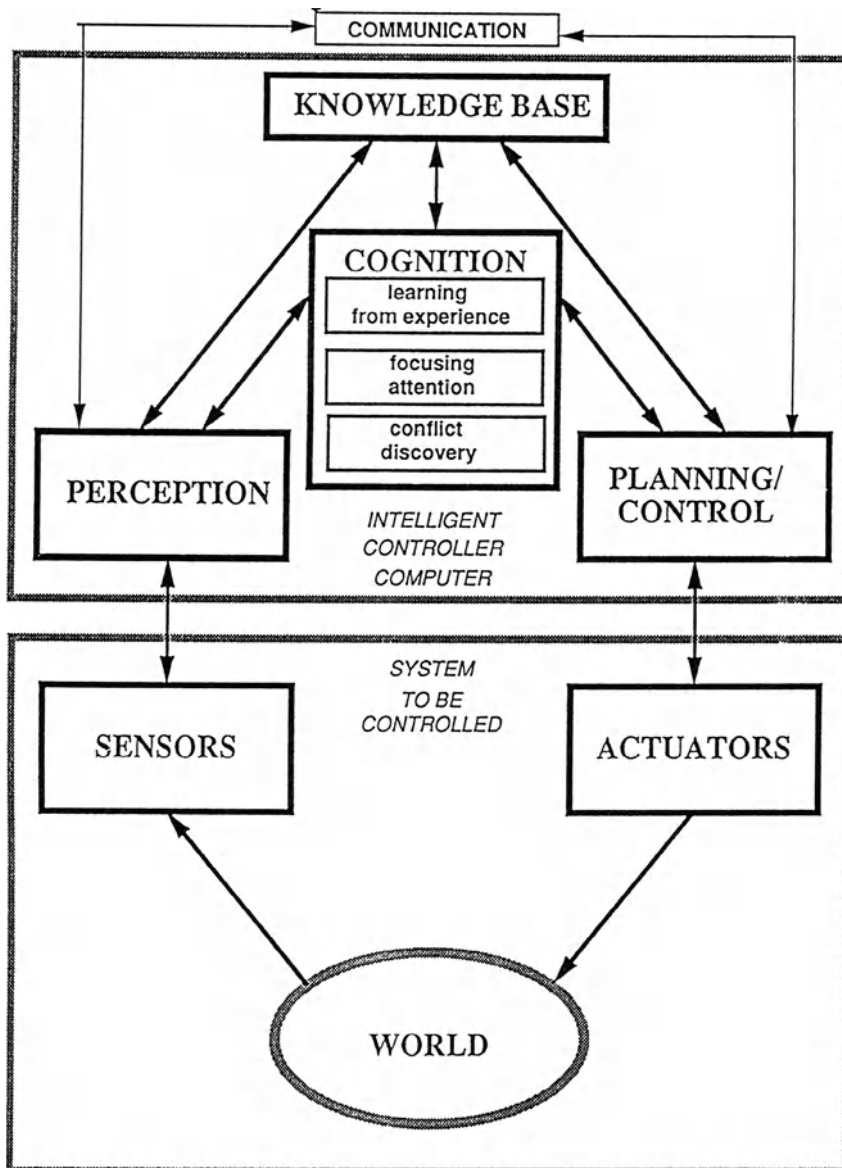


Figure 1. General Structure of the Intelligent Controller (or Intelligent Machine with a Controller).

can be a part of the team of similar intelligent machines. All teams constitute a “population”, and can be considered a specie.

A multiresolutional system can be introduced consisting of Specie(s)-Group(s)-Individual(s)-Organ(s)-Cell(s) hierarchy (SGIOC) which is driven by a corresponding hierarchy of goals. More technologically it can be formulated as a hierarchy of “kind of intelligent machines-teams of intelligent machines-intelligent machine-subsystem (sub-assembly) of an intelligent machine-components-primitive units”. We probably will need to talk about IC for all levels of SGIOC-hierarchy (global IC-center of a specie, team-IC, IC of an individual intelligent machine, decision making devices (IC-substitutes) of organs and cells). Clearly, in this context the mechanism of self-reproduction can be considered a regular issue for the area of intelligent machines.

The system shown in Figure 1 should be addressed in more detail. One can expect that the system of values plays a distinctive role as a part of knowledge base. We would assume that based upon perception of the results of its functioning, an Intelligent Machine develops a system of Values (an initial notion of goodness is presumed to be given to an Intelligent Machine at the stage of design). This process of values development is demonstrated in Figure 2 (see E-V triangle, where E stands for “Experience” and V stands for “Values”). On the other hand, the system of values together with Knowledge Base carves out from the base the sub-hierarchy of the Focus of Attention (A), and senses a conflict within the A -heterarchy (see the A-C triangle in Figure 2). The A-C-heterarchy is a good material for decision making activities. Thus, the decision making formations (DMF) can be introduced at corresponding levels of the SGIOC hierarchy and this will help to put in a proper prospective everything introduced later in the paper (more about SGIOC hierarchies see [60-63]).

One can expect that in SGIOC-hierarchy all levels have their particular language (for representation and information exchange). These languages form their hierarchy. Different languages might be expected horizontally and vertically. This is a necessary as well as sufficient condition because immediately after the language definition one can determine semantics, explain it as a set of rules and explain that these rules are to be stored for each level of the SGIOC-hierarchy and all storages of all levels together form a hierarchy of knowledge which in fact is a union of World Model (Representation) as well as the Intelligent Machine Model (WM+IMM). The following is assumed:

World Representation $\Longleftrightarrow$ Knowledge Representation
 (at any time instant) (at any time instant).

As soon as we consider representation as a time-dependent string of snapshots we can write that

World Representation(at an interval) $\Longleftrightarrow$ Event Representation.

The following issues should be addressed: a) the structure and the functioning of **decision making formations** (DMFs) which are rather important if not a major part of the **intelligent**

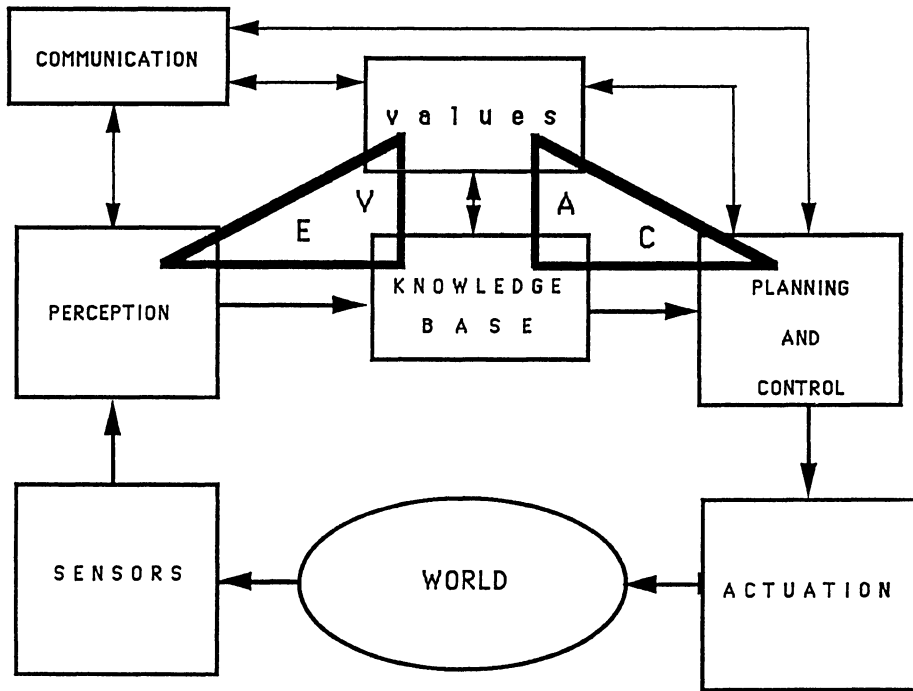


Figure 2. Experience-Values-Focus of Attention-Conflict Discovery in the structure of Intelligent Controller.

module (IM), b) means of communication without which IM cannot be shown as a component of the SGIOC-hierarchy , c) values generation which are required for proper decision making.

3. Knowledge Organization

3.1 Multiresolutional Knowledge Processing

A concept of nested hierarchical (*multiresolutional*, pyramidal) information (*knowledge processing*) (MRKP) is becoming increasingly important in the area of intelligent machines including robotics, computer vision, and knowledge-based material processing. *Multiresolutional Knowledge Representation is defined as the union of all monoresolutional representations.* The main idea of this concept is that the applicable model of a system cannot be built unless this system is considered simultaneously at several levels of resolution. Resolution is defined as a minimum volume of the state space that is distinguishable within a particular system of representation called *tessella* , and organization (discretization, quantization) of the state space is called *tesselation* if a particular size of tessella is being used efficiently as an element for building all descriptions of interest. A concurrent consideration of the system at several resolution levels is required, and the redundant representation is justified in which the "same" thing is represented several times with different resolution [1].

An example of the multiresolutional process description can be illustrated by a structure for a physical process shown in Figure 3. A definite technological process (say, metal casting, or assembling of a mechanical device) can be described with a definite resolution as a sequential/parallel network of subprocesses - phenomenological units (Ph₁ through Ph₅). If each of these phenomenological units is to be discussed at a higher resolution level (by using "sub-phenomenological units") it very well may happen that descriptions at this resolutional level "cannot talk to each other". Either, the levels of resolution do not match, or vocabularies are incompatible, or the physical processes under consideration belong to different domains, but the consistent model of the overall process cannot be delivered at a high enough level of resolution with a due consistency.

A notion of *multiresolutional knowledge representation* (MKR) is introduced for a variety of systems including data and/or knowledge bases, vision , control, and manufacturing systems, industrial automated robots, and (self-programmed) autonomous intelligent machines. Most of these applications are actually, or presumably utilizing intelligent modules with decision making capabilities, (or human operators performing similar functions). The structure of intelligent module is described in [6]. MKR is derived directly from the entity-relational representation of a system which is using a number of postulates of representation. Some of these postulates are establishing a graph representation for the system of interest which includes all levels of resolution since it contains not only the systems represented but also the nested system of their components. Another postulate presumes that the classes can be recognized among the multiplicity

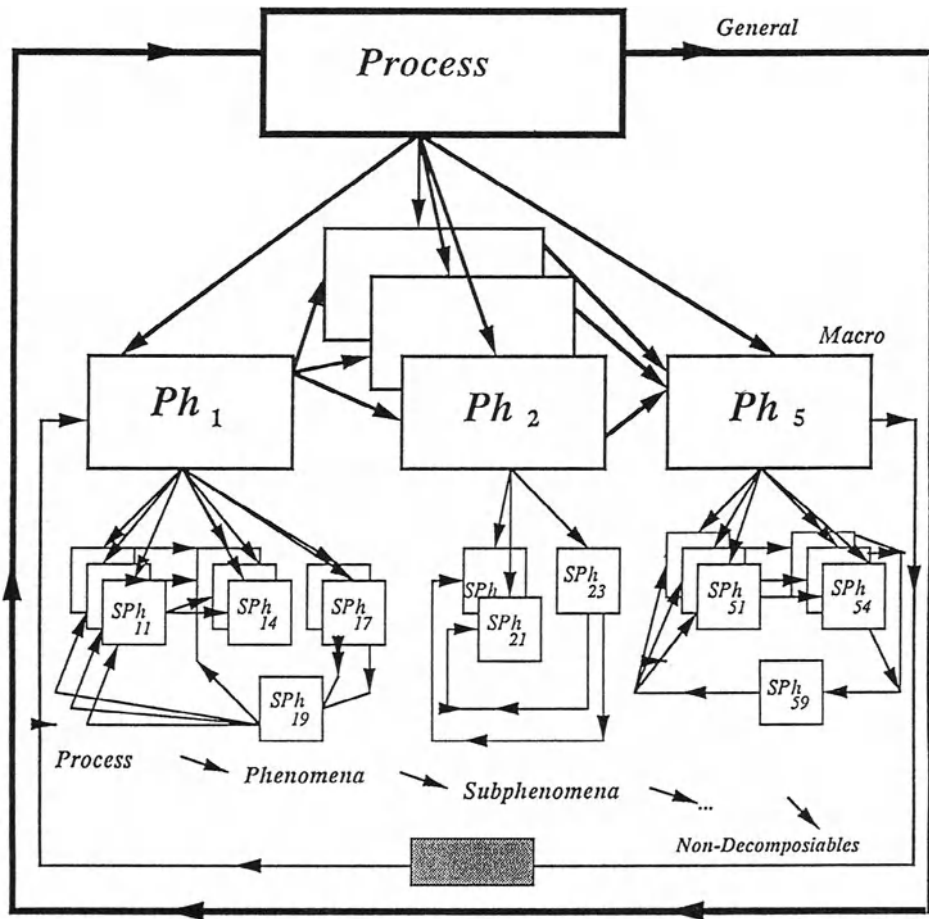


Figure 3. Phenomenological decomposition of a process

of labels, of those **commensurable** labels, i.e. belonging to the same space of consideration.

One can see that the structure can be visualized as a set of the interrelated scope graphs $\cup G_i$,

$\cup G_i R_{ij} G_j$; $i, j = I, II, III$, $i \neq j$, where R_{ij} is a relation among the elements of the graphs. Each of the scope graphs has a set of vertical (hierarchical) connections of the resolution levels and this set of connections is called a *hierarchy of the scope*. Within each level of resolution an entity-relational graph (*tessellatum*) exists which represents all entities and relations among them at a particular resolution (*tessella*). All tessellata belong to a particular hierarchy and are being considered together with it:

$$G_i = \bigcup_{ik} T_{ik} R T_i(k+1), \quad k=1, \dots, n, \quad i=I, II, III$$

(k is a number of resolution levels).

Each of the is unifying the set of inclusions for the tessellata

$$G_i = \bigcup_g [T_{ik} \supset T_i(k-1) \supset T_i(k-2) \supset \dots \supset T_i(k-n)]$$

where the inclusions are meant to represent the relations **R**. Any tessellatum of the higher resolution level can be transformed into the tessellatum of the lower level via mechanism of *generalization (abstraction)*. A set of all hierarchies with all tessellata related to each of the hierarchies forms a *heterostructure* (see **D-structure** in [7]).

General paradigm of multiresolutional control systems is presented in [40]. Any intelligent module transforms (sometimes, irreversibly) the knowledge it deals with, and this transformation affects the subsequent computation processes, e.g. those of decision and control. Several types of knowledge transformation are reviewed. One of them called knowledge filtering (KF) can be characterized by its volume and rate. The detrimental effect of KF can be compensated by the corresponding level of knowledge redundancy (and by the subsequent redundancy of decision making processes, followed by the action redundancies as well).

MKR allows for coding the system as a whole and not as a result of selecting only its limited subset. This allows for a harmonious control of a system. In [39] an example is described of using MRKP system for intelligent control of the OSPREY process in the metallurgy. Another system is now in the process of development for a plasma deposition machine.

A structure of GMKP operates as follows. It is presumed that this sub-object (SO) is a part of an object, which in turn, is a part of a particular Domain, which finally, is a part of the World.

Information concerned with SO (ISO) is obtained through the set of available sensors. The Sensor Information Carrier (SIC) delivers ISO to the system for MRKP in a form that contains information about the *code* carried by this particular SIC, and about the *modality* of this particular sensor. The code contains the information of the label and the value, this information should be

decoded, and the process of inference is performed, after which all information is structured and stored. The process of labeling and storing is actually the process of attachment (putting in correspondence) of the newly arrived information with the bulk of knowledge previously stored and verified. This process makes it *knowledge*.

As soon as the modality of sensor information is becoming known, a particular Domain of the World Knowledge is being evoked, and the mechanism of interpretation is being prepared taking in account the context, and listing the available rules that can be utilized by the system for dealing with the decoded information. New rules for interpretation can be obtained which in fact can affect the process of interpretation and inference and change the prior (recent) results. SO generates all sources of uncertainty: error of measurement (E), uncertainty of incompleteness (I), and uncertainty of redundancy (R). New EIR-uncertainties are generated within the code as a result of coding and communication; within the interpretation as a result of the EIR-interpretation properties, and within the storage as a result of EIR-properties of classification and other tools of information organization. All these factors should be taken in account when the degrees of belief are being determined. Usually they are generated within the loop of "learning - interpretation - storage".

3.2 Overview of the Situation in the Area of MRKP

MKR and associated techniques of MRKP was rapidly developing during past two decades from three different views: hardware MKR, visual images MKR, and algorithms MKR (with fuzzy boundaries). Using effectively multilevel, multilanguage structure of a computer is possible only if this multilevel structure is constructed by methods of aggregation (generalization, abstraction) and decomposition (instantiation) [8,9]. This area is linked with the problem of partitioning systems in order to achieve maximum of efficiency. Proper distribution of resolution among subsystems should provide the best utilization of equipment [6,10].

Another MKR problem adjacent to the problem of hardware partitioning was the following: how to partition something that has not been previously assembled, (e.g. partitioning of a curve) [16]. It was determined that the following factors must be taken in account: digitization and/or resolution of representation on hand, existence of multiple "views", and the set of attributes utilizable for describing the object to be partitioned. Linkage of all these approaches is undeniable to the "frame approach" from AI, and aggregation/decomposition methodologies of the earlier scientists belonging to the school of thought of General Systems Theory (e.g. see [12]). A method of multiresolutional curve representation is presented in [13]. Pyramid theories of image processing and interpretation have been promulgated during the last two decades in a multiplicity of well known books and papers by L. Uhr, E. Riseman, A. Hansen, S. Tanimoto, T.Pavlidis, M.Levine, R.Bajcsy, P.Burt, A.Rosenfeld [14-19]. Decomposition of entities is determined by the focus of attention at the level as illustrated in this sequence:

level of resolution $\Rightarrow$ context $\Rightarrow$ detail (tessella) $\Rightarrow$ focus of attention $\Rightarrow$ context
 $\Rightarrow$ next level of resolution

The well known quadtree structure [20] is not a multiresolutional structure in a sense that the accuracy of representation is the same at each level: the highest available accuracy of the level with the highest resolution. Recently, there was an attempt to fuzzify the upper levels images when the problem of planning was attempted using quadtree as a MKR system [21]. Truly MKR approach with using all tessellata for planning was successfully employed in [38]. It turned out that the set of hierarchical connections (those of G_i type) forms a "skeleton" that can be used as a good enough "syntactic" representation of various complicated shapes [23,24]. This phenomenon seems to have explanations within the principles of human perceptions reflected in the biological structure of vision system [25]. Multiresolutional representation turned out to be useful also for image segmentation and to region matching [26, 27].

MRKP is kindred to the fractal methodology of world representation [28]. Multiple-scale based approach to image representation and analysis [29] together with fractal-based techniques is actually application of the set of ideas characteristic for MKR. Here we are dealing with *simultaneous representation of all images at all resolutions* when the mechanism of generalization (or abstraction) is imposed upon the system by an external mathematical model. The last group of MRKP results is related to the multiresolutional algorithms. Somewhat interlaced with the fractal methodology are the algorithms of continued fractions [30,31]. Multiresolutional relaxation algorithms have been recommended for efficient dealing with texture [32]. A consistent and complete overview of the multigrid relaxation algorithms for image processing can be found in [33]. More detailed description of operations is given in [37].

3.3 Dealing with Unstructured Knowledge

One of the key problems in the area of intelligent control is dealing with the unstructured and/or hard to represent media. Even if the process variables are available, the model of this process is typically incomplete and fragmentary one. The output of the control process cannot be directly monitored, it allows only for a post-factum measurement and off-line learning. A research effort is undertaken at Drexel University attempting to provide the following contribution to the theory of intelligent control: an advanced architecture of the controller is developed based upon a new principle of knowledge inverse⁴ with an original combination of feedforward and feedback control principles using a multiresolutional state representation.

A new method of knowledge representation and modeling has been developed for the processes with incomplete and/or inadequate information support in which consistency of the model is being achieved via generation of the visual, acoustic, and thermal images blended with data from the conventional set of sensors. A computer vision system is being developed which is supposed to submit the lacking information for the consistent model of the process; this system is based on a novel principle of multiresolutional image processing in which conventional procedures of computer vision are replaced by a set of algorithms specifically proposed for dealing with unstructured and/or hard to represent media; a multiresolutional learning systems is being

⁴ Knowledge Inverse Operator in knowledge based controllers is similar to the systems inverse in a classical theory of linear systems.

developed with communication between the adjacent levels of resolution with the help of a hierarchical network of long-term learning and short-term adaptive channels which provide constant adjustment of knowledge inverse operator as well as compensator for parameters deviation [39, 57, 58].

Proper conditions should be satisfied for organization and processing of redundant information (knowledge) in the multiresolutional systems. Providing a sufficient degree of redundancy is one of these conditions. A definite set of rules of incorporating the redundant information (knowledge) must be applied for the system proper functioning. The significance of theoretically explainable techniques of dealing with redundancy of information (knowledge) is often overlooked. Several operators are discussed in [6] implicitly using redundancy of information (knowledge): generalization (abstraction), focusing of attention, etc. Phenomena of multiresolutional redundant perceptual organization are linked with the phenomena of error propagation (see [36]).

3.4 Uncertainty Generation and Propagation in MKR and MRKP

Multiresolutional system of Knowledge Representation (MKR) and processing (MRKP) is based upon postulates formulated in [39]. All of these postulates establish representation as a body which must be uncertain. Indeed, after the alternatives of the future decision are constructed (whether in the problems of design, or in the planning/control problems) these alternatives are to be compared. Consistent comparison can be done only if the judgment is developed about the uncertainty of the evaluation of our alternatives. The set of alternatives with a definite probability of occurrence, is obtained presumably by combinatorial methods discussed in ATG area [34,35].

The 6 famous Kolmogorov's axioms [41] are a mechanism for making judgment on the alternatives of decision. We will question the validity of the Kolmogorov's Axioms of 5 and 6 for the case of MKR. Indeed, the condition $\prod A_n = 0$ means that the events (sets) under consideration are incompatible. However, the infinite inclusion $A_1 \supset A_2 \supset \dots \supset A_v \supset \dots$ does not require necessarily that $\lim \Pi(A_n) = 0$, if $n \rightarrow \infty$. Everything depends on interpretation of inclusion. This becomes especially important when the process of consecutive generalization is considered.

We are addressing these questions by offering a general approach for dealing with processes of error propagation in the system, and by recommending measures of its reduction. We show that no judgment of error can be made before the system is organized as a hierarchy of generalizations. Thus, the hierarchy of resolution conscious information should be sought for from the SO. Then, the Code which arrives should be considered a generalized code which allows for nested hierarchical treatment (recursive interpretation, and/or consecutive refinement).

Now, the storage is becoming a multiresolutional system, and the whole right side of the structure is being adjusted with methodology of [6]: the source of knowledge is being treated as a multiresolutional structure, rules constitute a hierarchy of classes and a hierarchy of rules within the class, finally, the processes of learning are done consecutively with gradual involvement of each consecutive tessellatum. Then the following conceptual structure is required to support the

MRKP system in the view of dealing with processes of error generation and its reduction. The whole processing is considered as a multiresolutional system of consecutive encoding/decoding procedures. In a number of cases a hierarchy of sensors can be expected that makes the encoding subsystem working with a multiplicity of inputs to all levels.

4. Mechanisms of Decision Making: Task Generation

4.1 Synthesis of Decisions, and Decision Making

The structure of the Intelligent Controller can be visualized now as shown in Figure 4. It repeats the diagram shown in Figures 1 and 2 with the difference that multiresolutional organization of subsystems of Perception, Knowledge Base, and Decision Making (Planning/Control) is demonstrated explicitly. (One can see that we consider planning/control continuum to be the result of the operation of the subsystem of decision making. Plan is defined as a set of generalized activities to be performed to achieve a particular goal. Plan can be refined and the activities to be performed can be described in more detail. This process of refinement can be repeated many times until for the whole plan (or for the part of it which is well supported by available information) description of activities required can be made as precise as necessary. This precise assignment is considered to be a reference trajectory, and it allows for using it in the execution controller (see [6] and Figure 5). Structure from Figure 4 implies using three information heterarchies: H_G , H_S , and H_T which denote heterarchies of goal⁵, current state and task correspondingly.

Then the couple (H_G, H_S) constitute the problem to be solved and H_T is a solution for the problem. The planning/control algorithm (P/C) is supposed to produce a transformation

$$P/C:(H_G, H_S) \rightarrow H_T \quad (1)$$

The hierarchy of task decomposition is never given: it should be generated by the controller. How? Yes, the plan (a set of tasks at a level) forms a solution for a problem formulated as a task of the upper level. But where all these solutions are coming from? In other words, what is the mechanism of (1) that can be recommended throughout the NICS hierarchy?

The core of operation of the intelligent module in its part that is dealing with the goal heterarchy, does not differ from the process of the *automated theory generation* (ATG). The latter was first tackled in [34] and then was furtherly developed in [35] and other works. It is important to emphasize that any process of *representation is based upon theory generation*. Like in ATG, the subsystem of representation is supposed to synthesize a consistent system of tessellata constructed at different resolutions and transformable one into another. As it was mentioned in [40], the core of MRKP operations does not differ from the process of the automated theory generation (ATG) [34,35]. Since the mechanisms of generalization are involved, then *any process of representation is based upon theory generation*.

⁵ Goal is commonly understood as the state to be achieved. Thus H_G and H_S are both state descriptions: the desirable one, and the current one.

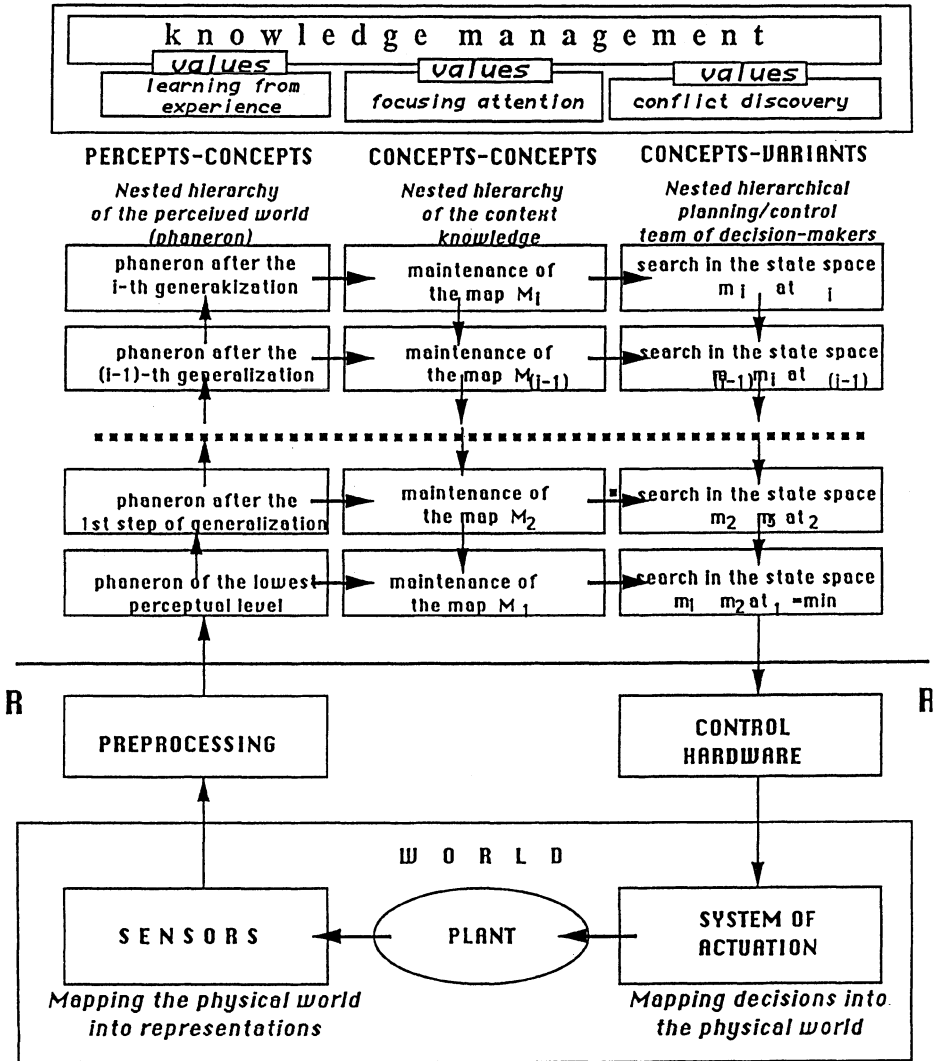


Figure 4. Multiresolutional Intelligent Controller

Existence of vocabularies for tasks at all levels is tacitly presumed. In the meantime, these vocabularies usually exist only partially. The need for a mechanism that is synthesizing the new words for the vocabularies of tasks was understood by J. Albus long ago and presented as a *list-processing mechanism*, or *cerebellar controller* [49-52]. Many researchers are using CMAC for learning purposes, however it remains what it was meant to be from the beginning: a conception which performs combinatorial synthesis on a set with particular properties.

Supplemented by CMAC (we will denote it C), expression (1) will be changed as follows:

$$P/C:(H_G, H_S) \rightarrow DM[C(H_G, H_S) \rightarrow \{H_T\}] \rightarrow \bigcup [H_T]_{\max} \quad (2)$$

x

where DM is an operator of decision making applied to the result of the combinatorial synthesis (see[54]),

$\{H_T\}$ -is a set of alternatives obtained at the output of CMAC,

$[H_T]_{\max}$ -the result of choice,

x-index of the hierarchy for which maximum solution is found, i.e. "m"

for the main hierarchy of decomposition, or "1,2,..." (where 1-a hierarchy of task decomposition which starts of the level under consideration and remains within this level, 2-a hierarchy of task decomposition for another upper level goal intersecting with the main hierarchy, 3-a hierarchy of task decomposition for one of the cooperative processes, etc.).

A single P/C at a level can be considered a Markov controller (e.g. as in [53]). For the hierarchies this process is described in [54]. It consists both of the C-operator for **synthesis** of the alternatives, and consecutive DM-operator for **choice**. C-operator has many incarnations in contemporary literature including CMAC [51, 52], Markov Controller [53], Conceptron [54], different neural-network based solutions, etc. Early GPS [55], and more recent SOAR [56] can be considered examples of possible C-operators for the upper (linguistic) levels of the NICS hierarchy of Planning/Control. Servo-levels often have C-operator too if search, or dynamic programming is applied in the Controller.

Practical conclusions from (2) can be listed as follows:

- Since each level of the multiresolutional system of representation has within-level (horizontal) relationships as well as branching new hierarchies and intersections, it would be improper to talk about hierarchies of representation, or hierarchies of task decomposition. Clearly in all of these cases we are dealing with **heterarchical representations**.

- **Three separate heterarchies should be maintained** for supporting the P/C operations: H_G , H_S , and H_T . The first two of them H_G , H_S are the state descriptions and can be associated with various aspects of the World Model from [6]. Both heterarchies can be maintained using the same system of representation. The third heterarchy H_T can be interpreted as transition function of the corresponding automata heterarchy. This heterarchy is usually expected to use a different representation system.

- It can be recommended to utilize a uniform approach to building vocabularies, grammars, and axioms for all levels of the multiresolutional heterarchies. *Every level is an object/task level*. At the lower levels (execution of commands) where analytical representation is very common (traditionally, and because of abundance of typical software packages), it is very easy to switch from analytical to automata representation, and then from automata representation to look-up tables which makes these levels uniform with the higher (linguistic) levels. This approach has been implemented for intelligent material processing (see [57, 58]).

4.2 Two Types of Task Decomposition

Task (command) is defined in [1] as an instruction

```
DO<Task>AFTER<Start Event>UNTIL<Goal Event>,
or
TASK COMMAND:= DO <TASK>
                WHEN (START EVENT)
                DO (TASK)
                UNTIL (GOAL EVENT)
                END-DO
```

One can see that tasks are controls. The statement DO<Task> is supposed to be understood by the performer: either by the lower resolution level which will DO the next decomposition of the TASK, or by the execution actuator which transforms the label of TASK into ACTION. In all of these cases the strings of tasks can be visualized as control laws. Control command can be applied if the state is known ("start event"), and when the final state is determined for this particular control command ("goal event"). So, it is clear that the hierarchy of TASK DECOMPOSITION (H_T) can be generated by the PLANNING/CONTROL SYSTEM if two additional hierarchies are given: a hierarchy of CURRENT STATE DECOMPOSITION (H_S) and a hierarchy of FINAL STATE, or GOAL DECOMPOSITION (H_G).

A set of n consecutive tasks (subtasks) for the i -th level represents a solution for the task T_{i-1} of the upper $(i-1)$ -th level

$$T_{i-1} \rightarrow P_i \rightarrow \{T_i(t_0, t_1); T_i(t_1, t_2); \dots; T_i(t_n, t_{n+1}); \dots; T_i(t_f-1, t_f)\}, 1 \leq n \leq f, \quad (3)$$

is called PLAN^{*}, or P_i to be performed at the i -th level when the task T_{i-1} should be executed at the $(i-1)$ -th level of the hierarchy of task decomposition. The task of the $(i-1)$ -th level under consideration T_{i-1} can generally be decomposed into a set of m parallel plans

$$T_{i-1} \rightarrow \{P_{i1}(t_0, t_f); P_{i2}(t_0, t_f); \dots; P_{im}(t_0, t_1)\} \quad (4).$$

* Thus, *plan* can be defined as follows: *plan* is a combination of *tasks* at the i -th level which are determined as a *solution* for the *problem* formulated as a *task* at the $(i-1)$ -th level of the *multiresolutional hierarchy* of a system.

Clearly, each of the plans in this set can be presented as (3). It looks that each of the (sub)tasks in (3) can be represented recursively using (3) and (4) which brings us to a complete recursive description of the overall task-decomposition hierarchy. In fact, it is not so. The following factors should be taken in consideration:

Two types of concurrent and consecutive plans. Two subsets of the concurrent plans can be considered:

a) plans which are being furtherly decomposed within the hierarchy under consideration (denoted P_{im} where i is the number of the level of resolution, and m stands for “main”), and b) plans which are not decomposed within *this* hierarchy. The latter, in turn can be divided in two groups: those decomposed as a part of intrinsic activities of the resolution level under consideration (denoted $P_{i=1}$ which means that for this branch of decomposition the i -th level of the main hierarchy is the 1-st level of the branching hierarchy), and those that can be decomposed only within another hierarchy which *intersects* the hierarchy under consideration⁷ (denoted P_{ix} where x should be selected depending on the particular name of the intersecting hierarchy).

Many examples of intersecting hierarchies can be found in different man-made and/or natural hierarchies (for example, see [47]).

In-level cooperation of concurrent plans. Parallel plans are talking to each other via cooperating tasks. Thus, in addition to tasks which can be represented in the form (1)-(3) there are tasks where factors of cooperation should be taken in consideration as follows:

DO<Task>AFTER<Start Event>UNTIL<Goal Event>

AND SIMULTANEOUSLY WITH<Parallel Event(s)>

or

```
TASK COMMAND:= DO <TASK>
                WHEN (START EVENT)
                SIMULTANEOUSLY WITH (Parallel Event(s))
                DO (TASK)
                UNTIL (GOAL EVENT)
                END-DO.
```

The sub-operator of cooperation can in turn be decomposed correspondingly⁸. This issue is addressed in [48].

Tasks belonging to different hierarchies simultaneously. This peculiar situation alludes not to obvious utilization of the same vocabulary for solving different problems within the same system but to the particular T_{ia} which is a solution for $T_{(i-1)a}$ in one hierarchy of task

⁷ The phenomenon of intersection of the task-hierarchies is a typical phenomenon in the systems of intelligent control. One can easily understand that the same task can have different cost-functional in different hierarchies.

⁸ Operator of cooperation should be considered in a broad sense, e.g. we can talk about “negative cooperation” in interaction with an adversary. It can be also unintended cooperation such as coupling. In [1] servos (see Figure 23) are communicating with each other, and this is not accounted for in a multiplicity of practical cases.

decomposition “a”, and to the particular T_{ib} which is a solution for $T_{(i-1)b}$ in another hierarchy of task decomposition “b” if $T_{ia} \leq T_{ib}$. This peculiarity affects the processes of decision making since, the solution can be selected out of the synthesized set of alternatives under two (or more) cost-functions simultaneously

5. Mechanisms of Decision Making: Structures for Goal Pursuit

5.1 Structure of the Control System

Most control applications are based upon a well understood model of the process (or “plant”) to be controlled. This model gives an opportunity to assign closed-form control algorithms which are to map control inputs to process outputs. Thus, in order to find the control inputs to be applied one should *invert* the required output using the model of the system which results in the required control inputs. Variations on this theme include adaptive inverse control approaches in which the controller continuously models the process in order to learn the inverse transfer function so that a desired response can be converted into the appropriate input values to produce that response.

Reference signal (trajectory) has been mentioned: this is the assignment which can be performed by the execution controller (level of the highest resolution). General structure of the execution controller is shown in Figure 5. The reference trajectory is introduced at the input of the “Open Loop Controller” which presumably should work as a “Plant inverse”. The same reference signal is utilized for comparing the output result with the assignment, and thus the feedback for error-compensation is initiated.

In cases where the process is nearly linear and/or dynamics of the system is relatively simple, when number of variables of interest is relatively small, and they are not coupled, these types of approaches can often produce satisfactory results.

However, when the process is dynamically sophisticated and/or non-linear, when system is MIMO and a coupled one, also when the process can be described only using differential equations in partial derivatives, then the result is that no computable high-resolution model is available and known adaptive techniques fail to converge. Finally, even when the system to be controlled is relatively simple but is not represented completely, it is very difficult to come up with a reliable control solution.

It is clear that what we definitely know about our future process we have to *preplan*, and what we are not sure about we have to *monitor* and *compensate*. Thus, the ideas emerge of combining planning (or off-line feedforward control) with feedback compensation.

The major concepts of compensation should be explored shown in Figure 6. Both concepts presume the feedforward “planner” which computes the input required for obtaining the desired output of the system. However, the concept “A” has feedback compensation applied after planner, and the concept “B” appeals to changes introduced at the planner’s input. This problem applies to level of IC.

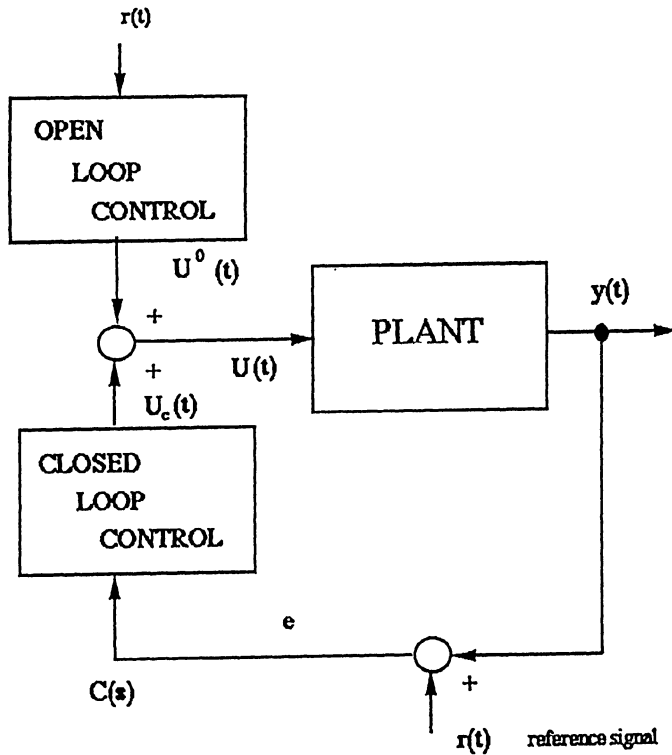


Figure 5. General structure of an Execution Controller.

OFF-LINE

input

assignment → PLANNER → • → SYSTEM → • → output

“A”



ON-LINE

input

assignment → • → PLANNER → • → SYSTEM → • → output

“B”



Figure 6. Two major concepts of compensation

All levels of the ISF are viewed as different implementations of the same concept, just solved differently. They are all compensating and computing a goal for the level below. There is search in all of them as well. In general however, the algorithms can be heterogeneous as well.

The following issues should be addressed:

- a possibility of on-line resolution control rather than having a set of preselected resolutions. It is especially important when the dynamic replanning is involved.

- a dilemma "continuous vs discrete" is not solved. With regard to moving from a discrete hierarchy of levels to a continuous hierarchical controller, one can note that the process is becoming continuous as accuracy of sensing and action degrades. Also, continuity occurs when the step size between levels gets very small (smaller than the accuracy of the levels to distinguish themselves).⁹

- general properties of a triplet of levels should be analyzed with respect to the input to a level (in terms of a goal for that level to achieve) and with respect to the output of that level (in terms of an assignment) are closely associated with the functioning of that level).

The complexity of situation is determined by the fact that planning requires lower resolution mostly symbolic representation, and feedback compensation presumes higher resolution analytical models. In these cases, the field of intelligent control provides a framework within which the control of the process can be achieved through the use of a hierarchical, nested, multiresolutional scheme (see [6, 7]).

A number of *ad hoc* approaches is known (for example [1] in control literature, and [2] in AI literature)¹⁰. These approaches illustrate that the problem of control design for this type of systems is far from being trivial one. The idea of nesting is looming in the view of experts [3]. The nontrivial part of it includes the relation among planning and feedback compensation.

5.2 Hierarchical Multiresolutional Control

To control a process which is not well understood and therefore not fully predictable¹¹, we need to move away from strategies based on the need for a complete, or single (that is, high) resolution model towards those based on knowledge which may take the form of rules of thumb, global relationships, and empirical results.

Knowledge of a complex system is not a collection of homogeneous models presented in a particular formal language: analytical or symbolic ones. In reality we are dealing with heterogeneous knowledge representation which includes whichever models available sometimes no

⁹ Any process formed as a sequence of a multiplicity of discretely is continuous if considered at a low enough accuracy.

¹⁰ Note the terminological difference between AI and Control Theory. AI differentiates between planning and reactive control, where Control Theory calls all "reactive" types of control "(feedback) compensation": planning is off-line reactive, and feedback compensation is on-line reactive. This would even cover the regeneration of planned trajectories (replanning) based on new information which becomes available in all kinds of systems down to the simplest PID algorithm.

¹¹ We believe that intelligent controllers are supposed to deal with this type of systems: which do not have an adequate representation system (because of incomplete knowledge, and/or hard to compute model).

models at all - just collections of expert statements. This type of heterogeneous knowledge bases lends itself readily to feedforward predictive planning [8]. However, the chief limitation of predictive planning is based upon what appears to be a universal relationship between expected error and the extent of time over which the prediction is made (on evolution of uncertainties see [9]).

We should also note that our *tolerance* of error versus prediction horizon is a similar curve because our overall reason for predictive planning in the first place is generally less affected by what we see far down the road than by that which is imminent. With this in mind, it is evident that an appropriate strategy would be to continue to plan for the long term while making the necessary adjustments in the short term. This then becomes *dynamic* predictive planning, augmented by a simplified scheme of reacting to current information¹² while staying within the boundaries set up by the planner.

Out of this line of reasoning based on dealing with heterogeneous knowledge representations (empirical, linguistic, and analytical) and the need for a “nested” control scheme (planning and compensation) comes an overall hierarchical control architecture which allows for different knowledge representation structures at each level of resolution. It would also allow the appropriate control algorithms to be used in conjunction with the available knowledge at the different levels of the hierarchy.

In a hierarchical control system, it is convenient to consider “triples” of levels. Triples are useful because the input to a given i -th level comes in as an assignment from a lower resolution $(i+1)$ -th level (level above) and it is interpreted as the goal for the execution of the i -th level which is typically making decisions at a higher rate than the level above. Once a decision is made at the i -th level the outgoing assignment is prepared for the subsequent $(i-1)$ -th level (level below). On the other hand, the lower $(i-1)$ -th level which receives decisions from the i -th level, is a source of updating information which is coming from the $(i-1)$ -th level. In a similar way the i -th level, is a source of updating information for the (i) -th level. Functionally these three adjacent levels can be considered levels for a) **planning** of the process, b) **navigation and guidance** of the process, and c) **execution** of the process¹³. On the other hand, if we add the $(i-2)$ -th level, then a triple can be considered which includes the following three levels:

i -th, $(i-1)$ -th, and $(i-2)$ -th.

Then, what was considered guidance for the previous triple (in its i -th level is becoming a predictive planning, the execution controller from the $(i-1)$ -th level is being considered now a guidance level, and the new $(i-2)$ -th level, or the lowest level of the new triple is its execution controller.

So, in each triple the planner delivers a trajectory to follow. (If this level is considered a

¹² This new information (from sensors) is supposed to serve for the processes of updating the World Representation. However, one should remember that this updating information is also imperfect: a) sensors have their errors and limitations, and b) some variables just cannot be perceived. For example, in the ISF machine thickness of the liquid layer (mushy zone) cannot be measured in principle, temperature of surface is hard to measure, surface roughness can be measured but is not easy to interpret, and so on.

¹³ If the number of actuators is more than 1, then the need in coordination is becoming inherent in the control levels [11].

middle level of the triplet, then it guides the system in such a way as to avoid deviations from the initially preplanned trajectory). Any real circumstances that create deviation from the preplanned trajectory can be treated in two different ways.

On the one hand, we can build a physical and/or mathematical model of these circumstances, and then try to plan how to return back to the initial planned trajectory. On another hand we can take in account our understanding that in time the validity of plan is diminishing (since the validity of judgments is reduced about the events remote in time [53]). So, probably we should constantly replan (repeat operation of planning) as new information is available. Then, it seems reasonable to make just a rough effort to compensate, and after this, to replan under the new set of circumstances. Thus, compensation should not be too precise, it should be fuzzy. This is why we recommend for compensation at all levels a fuzzy logic controller which can be considered a PID discrete controller¹⁴. Execution control is done in practice (often) by PID continuous controllers too [57, 58].

The following consideration should be taken in account in intelligent controllers design:

1) In MIMO controller, different inputs, outputs, and measurements belong to different resolution levels and often play role of mutual consecutive refinement.

Control is dealt with throughout the control system at increasing levels of resolution (top down) and thus, with increasingly more stringent requirements on rate of change and accuracy from level to level.

2) The same set of sensors provides information to different levels of resolution. The sensor information contains different types of information which can be utilized at different levels of resolution as required by the controller. Often the same sensor information is used by more than one layer, typically aiding a layer of lower resolution. In the most general case, all sensor information used by the hierarchical controller would pass through an interpretation module responsible for resolving conflicts, eliminating noise and parsing the information into the different levels of resolution for continuation on to the appropriate level in the hierarchy.

3) Granularity of the control level.

In this type of hierarchical architecture, the absolute value of the total volume of the state space of concern is much larger at the lower resolution levels than at the higher resolution levels. However, the time associated with traversing this space is also much larger. This means that the “total state space volume/total time span” ratio probably does not differ too much from level to level. It may be reasonable also to consider a “unit of space/unit of time” ratio in setting up the different levels of the hierarchy to insure an appropriate interface from one level to the next - too great a jump may hide an important level of abstraction which could be used to improve control over the process¹⁵.

¹⁴ One can easily verify that any PID controller is an analytical fuzzy logic controller. Indeed, in PID we assume that absolute values of deviations (of variable, its cumulative value, and its rate of change) should be added and the system should be compensated proportionally to this sum. Is not it what we are doing in fuzzy logic controllers assuming the linear change of the memberships, and then adding the preferences collected?

4) **Control priorities.** The hierarchical approach helps the designer to deal with the concept of priorities of control.

5.3 Process of Coordination

The progress in technology in recent years can be characterized by the increase of interest to the problem of coordination [43-45]. Two types of coordination are analyzed in this paper: inter-level (ILC) , and within-level coordination (WLC). ILC is performed by distribution of information and decision making activities among the levels of resolution in such a way as to minimize the cost-functional (in our examples we consider two cost functionals: time of computation and time of operation). ILC does not necessarily require a subsystem of coordination. WLC is dealing with multiple decision making agents at a level whose decisions affect each other and eventually affect the overall performance of the system. WLC makes the decision making processes independent of each other at each moment of time. An *intentionality map* is being developed and maintained by the subsystem of WLC for each of the decision making agent. A computer architecture of a nested hierarchical controller is introduced for a hierarchical multi-actuator system.

In such a system each of the actuators is working in its own state space and has a) its own pseudo-goal, as well as b) its own system of constraints. Both position of the goal and configuration of constraints must be constantly recomputed under supervision of Decoupler and Coordinator. Then the FLDT driven controller computes the system of jerks, accelerations, and speeds to be followed in order to provide minimum time operation (also bold lines on time diagrams). The algorithm of coordination works in the following steps.

As soon as the motion started, Coordinator recomputes the configuration of the constraint boundary taking in account the information about motion of all actuators of the system. It is done by creating the "Intentionality Map" for each of the actuators. Planning/control subsystem replans the path for the new configuration of constraints. Then Decoupler recomputes the switching times, and the motion continues in a little bit different direction. This process is repeating each discrete of time accepted in the system. As a result the trajectory of real motion has a configuration which under no circumstances could be computed in the beginning unless the process of negotiations would not be provided in a comprehensive way. The problem is not yet solved concerning the degree of optimality in the set of trajectories obtained in this way.

The Knowledge Based Organizer, and the Dispatcher both are submitting information to the General Coordinator which provides joint consistent operation of the set of N Coordinators. Each of them coordinates activities of a set of m subsystems consisting of controllers, actuators, and sensors. Each elementary closed-loop subsystem of controller-actuator-sensor equips a particular elementary process which can be characterized by a relative local independence. Sensors provide

¹⁵ Here we are talking actually about so called "granularity" of the state space which determines our ability to perform certain search procedures. Since most of the search algorithms are NP-complete, their practical application is feasible only if "granularity" of the search space is selected properly and the system is computable.

information for closing the loop of the subsystem via system of Sensor Integration related to the particular Coordinator, and providing the main feedback for this Coordinator. A definite specifics should be mentioned of the coordination processes in the systems with decoupled subsystems (e.g. using feedback linearization, and decoupling, or FLDT as in [42]). Decoupled subsystems can be viewed as independent decision makers [6,7].

The procedures of the trajectory generation are based upon the concept of open loop controller. The desirable trajectory is planned based upon the principles of optimum control, and then it is inverted in order to find the input which is required to provide the optimum motion. Since the optimum trajectory is computed based upon definite assumptions the real values of physical parameters lead to deviations of real trajectory from the prescribed plan. Thus, the feedback control applied in this system is dedicated to only compensate the deviations from the prescribed plan. The concept of minimum time control for a system with FLDT leads to a particular structure of a program: all compensations are supposed to be done by corrections in switching times.

All Sensor Integrators submit their information to the General Sensor Integrator which serves as a feedback for the General Coordinator [39]. Information from sensors is being processed in a multiresolutional fashion as described in [6,7,40] interacting with the domain and context knowledge within the multiresolutional hierarchy of knowledge, and generates plans and controls within the similar p/c subsystem. Using principles demonstrated in [42] for a multilink manipulator, we apply method of generalized transformation for developing a system of linearization and decoupling.

5.4 NICS-operator

General structure of the Intelligent Controller is known as a set of concentric feedback loops [7,56]. Each feedback loop is declared for a particular resolution level. Analysis of these loops is becoming possible if the system of PLANT representation will be built as a mirror reflection of the Nested Intelligent Control System. In *Figure 7*, a concept is illustrated which puts in correspondence resolution levels of NICS and PLANT by forming concentric loops: one loop per resolution level.

Each loop works with its own time-scale. The loops with higher resolution are dealing with fast control of processes with small time-discrete while the loops with lower resolution concentrate upon slower processes and have larger time scale. The time related matters can be treated as recommended in [6].

This gives an opportunity to focus on providing satisfaction of consistency relations among all resolution levels within particular subsets of information corresponding to heterarchies of perception, global knowledge representation, H_G , H_S , and H_T . Then a unified set of procedures can be applied recursively: NICS-operator.

In this Section two issues are addressed: of consistency, and of recursiveness of NICS-operator.

Consistency of NICS-operator

The operation of task-decomposition requires all tasks to be nested in a form of (2). Note, that the *rule of nesting* is the main property of the NICS-operator, it is a *law of NICS architecture*. This law does not depend on the form of representation (linguistic, or analytical), independently of the vocabularies utilized at particular levels, *the tasks must be nested*. It is easy to demonstrate that satisfaction of this property implies nesting of the vocabularies, grammars, and axioms of the adjacent levels. Moreover, solutions are supposed also to be nested which follows the property of nesting of plans (3, 4).

These properties can be implemented as *consistency checks*. The system is consistent if these properties hold. But if they hold, the levels cannot be treated independently, the results of all transformations should be verified by applying the consistency check. Thus, some results that might be considered optimal for a level, (and indeed are contributing to the minimization of a particular cost functional selected for a particular system), these results should be rejected if consistency check is not satisfied.

In practice this entails a rule: *NICS must be designed as a whole*¹⁶: in this case the consistency can be checked. One cannot postpone design of the level (i-1) until the time after the design of the i-th level is fully completed, the level is tested and even manufactured. It can happen that after the i-th level is designed the overall system will be inconsistent, or the requirement of consistency will require compromise the conditions of optimality, or the constraints of the system.

Recursiveness of NICS-operator

We propose to treat all multiplicity of operations described above as a single operator: NICS-operator with inputs, outputs, states, transition and output functions determined according to heterarchical representations mentioned above. This allows for development of unique and potentially beneficial computer architectures and methods of computational analysis of Intelligent Controllers based on NICS-operators.

This will open an opportunity to develop a special computer component base which can become crucial for development of Intelligent Controllers based on NICS architecture.

6. Conclusions

6.1 Application

We have an experience of applying the Intelligent Controller methodology for two applications: autonomous mobile robots [6, 38], and intelligent material processing [39, 57, 58]. It is important to watch how the new methodology works with different professionals which are part of the Intelligent Controller team. Accepting a particular control logic does not seem to be a trivial

¹⁶ It is a commonplace that designers underestimate the significance of the consistency check (or are forced by circumstances to neglect the rule of designing NICS as a whole). Of course, the requirement of designing NICS as a whole seems to be overly unconventional, and one can expect that time and patience will be required until this requirement will be incorporated in the practice of systems design.

problem. Let us see how it looks like in the application for spray casting [57, 58].

a) *Extremely low resolution. Off-line operation. Extremely slow activities - constitute the framework for the Planner Model.* The metallurgist determines that the goal of quality can best be achieved by controlling microstructure, productivity, and accuracy of complex shape manufacturing and passes that assignment to the Planner part of the Intelligent Controller. It uses information obtained as a result of applying off-line a microscope, and mechanical instruments for measuring dimensions of the preform, as well as prior evaluations of the amount of material wasted. These results are the basis for development of the general phenomenological model of the process utilized at this stage of Intelligent Control.

b) *Very low resolution. Off-line operation. Very slow activities - are utilized for Planner Model refinement.* The materials-scientist builds a theoretical model which testifies that good microstructure is formed by the control of the solid/liquid fraction at the surface of the preform. The latter is determined by the rate of delivery of the material and heat to the preform (as well as subsequent heat dissipation), which in turn is determined by the distribution of speeds, masses, and temperatures of the droplets in the spray. Sensors cannot be used to verify this model, it is a theoretical model of the process.

c) *All spectrum of resolutions are considered at this stage. Off-line operation. Slow activities - this is the basis for Plan refinement and "Navigation" of the Process.* The engineer is trying to find how to control these distributions of the speeds, masses, and temperatures of the droplets by the values used in the model (spray height, substrate motion, secondary gas pressure, motion of the spray). He builds analytical and/or linguistic models (rules) of these relationships and passes his assignment to the Intelligent Controller (Open Loop Feedforward assignment) in the form of World Representation reasonable goal of the process, and specifications of what should be measured, when, and how. He equips the system with actual set of sensors which (he believes) are cost-effective.

d) *Low resolution. Off-line/on-line operation. Not too slow action - are the basis for deciding upon reference trajectories.* The Planner and the Assigner of the Open-Loop reference trajectory invert the Engineer's (Designer's) rough assignment into the set of control profiles. In such a complicated model he cannot do it mathematically, so he is searching in the state space. His sensors are: the results of measuring the surface roughness which has been determined to be a measure of the solid/liquid fraction, as well as the prediction (obtained by off-line, or on-line simulation) of what is the thickness of the liquid layer.

e) *Medium resolution. On-line application. . . Medium fast action--is the basis for Guidance of the process.* The fuzzy-logic knowledge based controller determines that the surface roughness testifies for a need in change of the solid/liquid fraction which can be affected (for example) by the metal flow rate and passes that assignment to the PID in charge of metal-flow rate. His sensor is a load cell and vision.

f) *High resolution. On-line application. Very fast action --are the basis for the execution control part of the fuzzy logic controller.* A PID uses crucible overpressure to control metal flow rate, so it uses both the load cell feedback and the overpressure measurement. It uses such sensors as the optical encoders on the shafts of the actuators.

With regard to using sensors at different levels of the hierarchy, note that in general/ideal

case, all sensors would pass through an "information representation" stage where conflicts are resolved, etc. Coming out of this stage, the information at the appropriate levels of resolution are passed to the different levels of the hierarchy.

The levels of resolution are algorithmically homogeneous but heterogeneous from an implementation standpoint. This can be seen if we compare the three levels of the existing examples of application: heuristic search, fuzzy logic control, and PID. Each is a compensation-style algorithm which is implemented in a different manner to best deal with varying requirements for accuracy and speed in their results.

6.2 System-theoretical View of Intelligent Machines

Modeling of the system with intelligent controllers is a prerequisite for the successful controller operation. However unclear the process at hand is it should be viewed as a mirror reflection of the controller to be applied (and vice versa). Thus the multiloop controller shown in Figure 7 should be transformed into a diagram Figure 8. The model of NASREM [1-4] therefore can be enhanced by

- a) inclusion of the model of object as a mirror reflection of the intelligent controller applied, and
- b) closing the loops with forming nested multiloop structure.

The overall model of NICS is shown in Figure 9. One can see that all three heterarchies: Perception, Knowledge and Decision Making are similarly interrelated in "reality" as well as in the domain of representation in intelligent controller -with the difference that the information flow is inverted.

Another important feature of NICS is that the heterarchy of Knowledge and the heterarchy of World (middle parts of the picture) are not connected directly, and the verification of representation (identification) can be done only via joint activities of Sensor-Perception and Decision Making-Actuation couples.

6.3 Future Directions

In this concluding section we discuss the research priorities in this area.

a. Continuous vs discrete controller. It may appear reasonable to think about ever finer differentiations between the levels of this form of controller until the accuracy of adjacent levels is no longer sufficient to distinguish between them. At this point the heterarchy controller should become continuous in nature rather than discrete, with a single analog value controlling the location within the spectrum of levels at which you wish to operate at any given time in the process. A mathematical description would include the idea of moving from a description of the levels of the controller based on difference equations to one based on differential equations. The exploration of this concept may also borrow significantly from the fixed point theorem in mathematics.

However, we expect that the assumption of "distributed heterarchy" will contradict the very goal of its introduction: to reduce the volume of computations under the limit of existing (available) computer power of the controller.

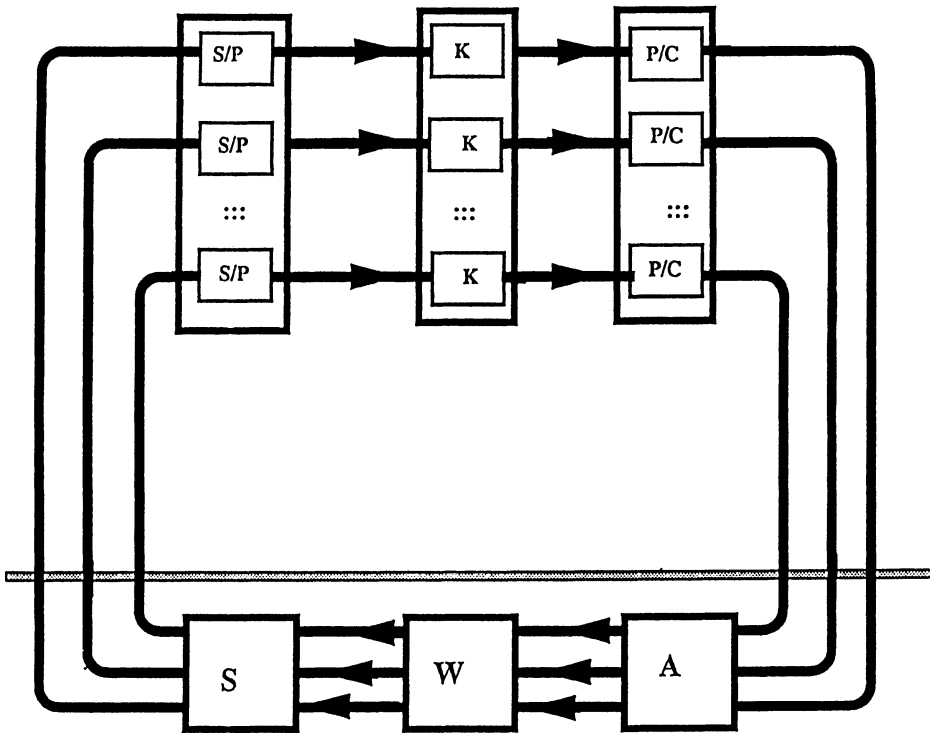


Figure. 7. Intelligent Controller as a set of concentric multiresolutional control loops.

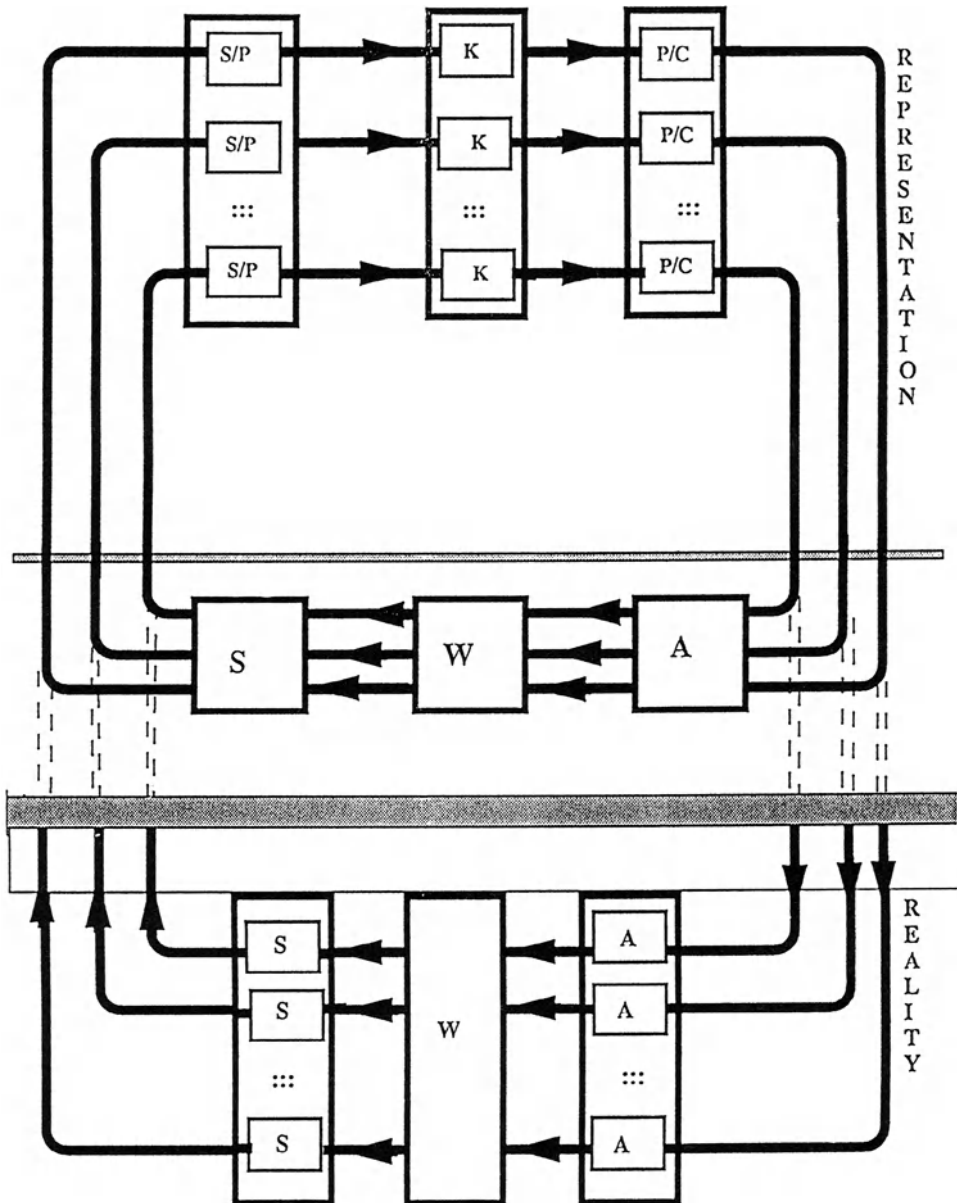


Figure 8. Symmetry of the multiloop Controller and the System to be controlled

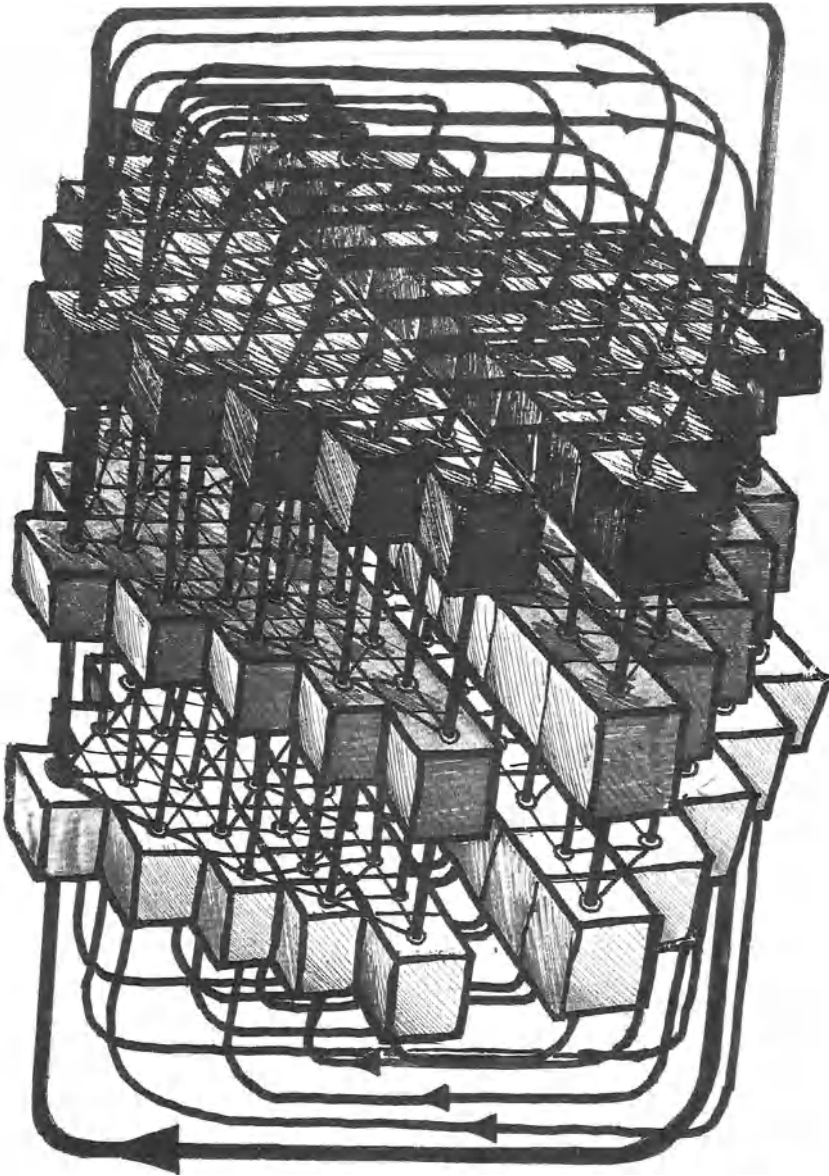


Figure 9:
Metaphorical representation
of NICS-concept

b. Algorithms to be applied. Although the existing examples of the NICS controller are useful for a discussion in this paper, they employ different implementations of the same basic feedback compensation-oriented control algorithm at each level. The implementations only vary in terms of their level of granularity complexity, with the finer granularities and higher complexities used at the lower resolution levels. Actually, the algorithms of control at all levels explore the same idea of fuzzy proportional compensation of deviation with or with no attention to the effectiveness of the compensation process and efficiency of the process of computation.¹⁷ Different types of control algorithms might be explored for use at the different levels of the hierarchy.

c. Learning in the IC.

-Learning at the level of Planner is based upon successes or failures of planning results based upon experience (or *post-factum*). When preplanning is becoming advantageous its results can be incorporated in the model utilized at the level of Planner.

-Fuzzy Logic Controller has similar needs. It operates in a local area of the state space. When it takes action, there is always a delay which can be accommodated by noting previous and current dynamic effects and predicting the future with a simple/local/linear model of the process. It is important to know how does this delay affect the operation. One can see that the the problem of this level is similar to the problem of Planner level.

Learning the new information, or updating the world representation is important for all levels of control. Both quantitative as well as conceptual learning subsystems should be explored.

d. IC-Shell. As the field of Intelligent Control matures and applications for hierarchical, multiresolutional control of processes with incomplete knowledge arise, there will be a demand for an economical way to develop these controllers without the need to completely build all components from scratch each time. It is our intention to construct an IC-Shell package specifically for this purpose.

Components for IC-Shell construction will include a model-independent¹⁸ and heuristically-guided search algorithm, a fuzzy logic controller (discrete symbolic PID) module, multiple continuous analytical PID-style controllers, advanced graphical user interfaces and a framework known as a "System Integration Toolkit" in which to pull it all together.

The SIT and the IC-Shell will be portable across a variety of popular microprocessor platforms and will support the parallel processing capabilities necessary for these types of systems. A growing family of sensors and associated electronic interfaces will be supported to allow for the required interaction with the process to be controlled.

SIT will include a number of tools which will IE: Any discrete process is continuous if considered at a low enough accuracy. It will allow to represent the system to be controlled in a form susceptible to the principle of operation of the IC-shell. This representation will be done in a heterogeneous knowledge-base fashion which will allow to use state-space search, provide the desired granularity at different resolution levels, allow symbolic representation at rough granularity, and analytical representation at fine granularity. (Actually, controllers of the different levels of

¹⁷ It turns out that this idea is a fundamental one and the possible technical improvements cannot change anything in the very principle of the overall approach.

¹⁸ Model is usually meant to represent the technological system and/or process.

resolution cannot be homogeneous due to the heterogeneous nature of the of the different types of knowledge associated with the different levels of control resolution, forcing different representations).

References

1. J. S. Albus, H. G. McCain, R. Lumia, *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*, NIST Technical Note 1235, April 1989 Edition (Superseded NBS Technical Note 1235 Issued July 1987), US Government Printing Office, Washington, 1989
2. J. S. Albus, *System Description and Design Architectures for Multiple Autonomous Undersea Vehicles*, NIST Technical Note 1251, September 1988 Edition, US Government Printing Office, Washington, 1988
3. K. Chaconas, M. Nashman, *Visual Perception Processing in a Hierarchical Control System: Level I*, NIST Technical Note 1260, June 1989 Edition, US Government Printing Office, Washington, 1989
4. J. S. Albus, R. Quintero, H.-M. Huang, M. Roche, *Mining Automation Real-Time Control System Architecture Standard Reference Model (NASREM)*, NIST Technical Note 1261, Vol.1, May 1989 Edition, US Government Printing Office, Washington, 1989
5. M. Mesarovic, D. Macko, Y. Takahara, *Hierarchical Multilevel Systems Theory*, Academic Press, NY, 1970
6. A. Meystel, "Theoretical Foundations of Planning and Navigation for Autonomous Robots", *International Journal of Intelligent Systems*, Vol. 2, No. 2, 1987, pp. 73-128
7. A. Meystel, "Intelligent Control in Robotics", *Journal of Robotic Systems*, 1988
8. B. Moszkowski, "A Temporal Logic for Multilevel Reasoning about Hardware", *Computer Magazine*, February, 1985, pp. 10-19
9. S. Y. H. Su, C. Huang, P. Y. K. Fu, "A New Multilevel Hardware Design Language and Translator", Proc. of 5-th Conference on Computer Hardware Description Languages, Kaiserslautern, West Germany, 1981, pp. 155-169
10. R. Agrawal, H. Jagadish, "Partitioning Techniques for Large-Grained Parallelism", *IEEE Transactions on Computers*, Vol. 37, No.12, 1988, pp. 1627-1634
11. M. Fischler, R. Bolles, "Perceptual Organization and Curve Partitioning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No.1, 1986, pp. 100-106
12. M. Mesarovic, "Foundations for a General Systems Theory", Ed. by M. Mesarovic, Views on General Systems Theory, Proc. of the 2-nd Systems Symposium at Case Institute of Technology, Wiley, NY 1964
13. W. Kropatsch, "Curve Representation in Multiple Resolutions", Proc. of the 8-th Int'l Conference on Pattern Recognition, Paris, France, October 27-31, 1986, pp. 1283-1285
14. L. Uhr, "Layered "Recognition Cone" Network That Preprocess, Classify, and Describe", *IEEE Transactions on Computers*, Vol.C-21, 1972, pp. 758-768
15. A.R. Hansen and E. M. Riseman, *Preprocessing Cones: A Computational Structure for Scene Analysis*, Technical Report of the Computer and Information Science Department, University of Massachusetts, 1974
16. S. Tanimoto, T. Pavlidis, "A Hierarchical Data Structure for Picture Processing", *Computer Graphics Image Processing*, No.4, 1975, pp. 104-119
17. M. D. Levine, J. Leemet, "A Method For Nonpurposive Picture Segmentation", Proc. of the 3-rd Int'l Joint Conference on Pattern Recognition, Coronado, CA 1976, pp. 494-498
18. D. Rosenthal, R. Bajcsy, "Visual and Conceptual Hierarchy - A Paradigm for Studies of Automated Generation of Recognition Strategies", Ed. A. Rosenfeld, Multiresolutional Image Processing and Analysis, Springer Verlag, Berlin, 1984, pp. 60-76

19. P. J. Burt, "The Pyramid as a Structure for Efficient Computation", Ed. A. Rosenfeld, *Multiresolutional Image Processing and Analysis*, Springer-Verlag, Berlin, 1984, pp. 6-36
20. H. Samet, "The Quadtree and Related Data Structures", *ACM Comput. Surv.*, Vol.16, No.2, June 1984
21. S. Kambhampati, L. Davis, "Multiresolutional Path Planning for Mobile Robots", *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 3, 1986
22. K.-S. Fu, "Syntactic Pattern Recognition", Eds. T. Y. Young, K.-S. Fu, *Handbook of Pattern Recognition and Image Processing*, Academic Press, Orlando FL, 1986, pp. 85-118
23. S. M. Pizer, W. R. Oliver, "Hierarchical Shape Description Via the Multiresolutional Symmetric axis Transform", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, 1987, pp. 505-511
24. A. Dill, M. Levine, P. Noble, "Multiple Resolution Skeletons", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, 1987, pp.495-504
25. J.-P. Crettez, S. Tanimoto, "Perceptual Constancy and the Multi-Layer Visual Model: Position and Size Invariance", *Proceedings of the Workshop on Computer Vision*, Washington DC, 1984, pp. 518-52.
26. W. Grosky, R. Jain, "A Pyramid Based Approach to Segmentation Applied to Region Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 5, 1986, pp. 639-650
27. R. Ohlander, K. Price, D. Raj Reddy, "Picture Segmentation Using a Recursive Region Splitting Method", *Computer Graphics and Image Processing*, Vol. 8, 1978, pp. 313-333
28. B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman and Co, NY 1983
29. F. Mokhtarian, A. Mackworth, "Scale-Based Descriptions and Recognition of Planar Curves in Two-Dimensional Shapes", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No.1, 1986, pp. 34-43
30. U. Cheng, "On the Continued Fraction and Berlekamp's Algorithm", *IEEE Transactions on Information Theory*, Vol. IT-30, No.3, 1984, pp. 541-544
31. N. Blachman, "The Continued Fraction as an Information Source", *IEEE Transactions on Information Theory*, Vol. IT-30, No.4, 1984, pp. 671-681
32. F. Cohen, D. Cooper, J. Silverman, E. Hinkle, "Simple Parallel and Relaxation Algorithms for Segmenting Textured Images Based on Noncausal Markovian Random Field Models", *Proc. of the 7-th Int'l Conference on Pattern Recognition*, Montreal, Canada, 1984, pp. 1104-1107
33. D. Terzopoulos, "Image Analysis Using Multigrid Relaxation Method", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 2, 1986, pp. 129-139
34. S. Amarel, In *Principles of Self-Organization*, Trans. of the University of Illinois Symposium on Self-Organization, June 8-9, 1961, Eds. H.Von Foerster, G. Zopf, Jr., Pergamon Press, Oxford, 1962
35. R. Davis, D. Lenat, *Knowledge-Based Systems in Artificial Intelligence*, McGraw Hill, NY 1982
36. A. Meystel, "Multiresolutional Models of Uncertainty Generation and Reduction", *Proceedings of the JPL Symposium on Telerobotics*, Pasadena, CA 1989
37. A. Meystel, G. Chen, *Multiresolutional Structure of Assembled Objects*, Technical Report, Drexel University, Philadelphia, PA 1989
38. P. Graglia, *Path Planning in the Multiresolutional Traversability Space*, MS Thesis, Drexel University, 1988
39. D. Apelian, A. Meystel, "Knowledge Based Control of Material Processing", *Proceedings of the 118-th TMS Annual Meeting*, February 27-March 2, 1989, Las Vegas, Nevada; Volume on Symposium "Metallurgical Processes for the Year 2000 and Beyond", Publ. by The MMM Society, 1989.
40. A.Meystel,"Techniques and Potential Capabilities of Multiresolutional Information (Knowledge) Processing" *Proceedings of the JPL Symposium on Telerobotics*, Pasadena, CA 1989
41. A. Kolmogorov, *Foundations of the Theory of Probability*, NY 1956

42. A. Meystel, A. Guez, "Optimum Positioning of a Servomechanism", Proc. of 21-st IEEE Conference on Decision and Control, Vol. 3, Orlando, FL., 1982
43. J. Fortuni-Amat, J. Talavage, "A solution Procedure for the Hierarchical Coordination of Constrained Optimizing System", Int. J. Systems Sci., Vol. 12, No. 2, 1981, pp. 133-146
44. E. Durfee, V. Lesser, D. Corkill, "Coherent Cooperation Among Communicating Problem Solvers", IEEE Transactions on Computers, Vol. C-36, No.11, 1987, pp. 1275-1291
45. Y. Bestaoui, "Adaptive Hierarchical Control for Robotic Manipulators", Robotics, Vol. 4, 1988, pp. 145-155
46. V. Hayward, S. Hayati, "KALI: An Environment for the Programming and Control of Cooperative Manipulators", Proc. of the 1988 ACC, Vol. 1, Atlanta, GA, 1988, pp.473-478
47. T. F. H. Allen, T. B. Starr, *Hierarchy: Perspectives for Ecological Complexity*, The University of Chicago Press, Chicago and London, 1982
48. A. Meystel "Coordination in a Hierarchical Multiactuator Controller", Proc. NASA, Conf. on Space Telerobotics, JPL Pasadena CA 1989
49. J. S. Albus, "Theoretical and Experimental Aspects of a Cerebellar Model", Ph.D. Dissertation, University of Maryland, December, 1972
50. J. S. Albus, "The Cerebellum: A Substrate for List-Processing in the Brain", in *Cybernetics, Artificial Intelligence, and Ecology*, ed. by H. W. Robinson, D. E. Knight (Proc. of the 4-th Annual Symposium of the American Society for Cybernetics), Spartan Books, New York, 1972
51. J. S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)", Trans. ASME, J. of Dynamic Systems, Measurements, and Control", Vol. 97, pp. 220-227, Sept. 1975
52. J. Albus, "Mechanisms of Planning and Problem Solving in the Brain", *Journal of Mathematical Biosciences*, Vol. 45, pp. 247-293, 1979
53. A. Belostotsky, A. Meystel, "Hierarchical Parallel Search" *Journal of Intelligent and Robotic Systems*
54. A. Meystel, "Purposeful arrangements via search in the version space," Proc. IEEE Int'l Conf. on Systems, Man and Cybernetics, Halifax, Nova Scotia, 1984
55. A. Newell, H. A. Simon, "GPS, A Program that Simulates Human Thought", Eds. E. A. Feigenbaum, J. Feldman, *Computer and Thought*, McGraw-Hill, NY, 1963
56. A. Newell, *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA 1990
57. B. Cleveland, A. Meystel, "Dynamic Predictive Planning + Fuzzy Logic Control= Intelligent Control", Proceedings of the 5-th IEEE Int'l Symposium on Intelligent Control, Philadelphia, PA 1990
58. A. Meystel, et al., "State Space Search for an Optimal Trajectory: Planning for the ISF Metal Processing Control System", Proceedings of the 5-th IEEE Int'l Symposium on Intelligent Control, Philadelphia, PA 1990
59. A. Meystel, "Intelligent motion control in the anthropomorphic machines," in: *Applied Artificial Intelligence*; pp. 317-352, editor, S. Andriole, Petrocelli Books, Princeton, NJ 1985
60. J. S. Albus, "Outline for a Theory of Intelligence", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 21, No. 3, May/June 1991
61. J. S. Albus, *Brains, Behavior, and Robotics*, BYTE/McGraw-Hill, Peterborough, N.H., 1981
62. T. F. H. Allen, T. B. Starr, *Hierarchy: Perspectives for Ecological Complexity*, The University of Chicago Press, Chicago, 1982
63. G. N. Saridis, "Intelligent Robotic Control", *IEEE Trans. on Automatic Control*, Vol. AC-28, No.5, 1983

Control of Contact in Robots and Biological Systems

Neville Hogan

Massachusetts Institute of Technology, Departments of Mechanical Engineering and Brain and Cognitive Sciences, Room 3-449D, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA

Abstract

One fundamental aspect of manipulation will be discussed to illustrate how a comparative study of robotics and biological movement control can yield deeper insights into both fields. Mechanical interaction with objects is a prerequisite for manipulation, yet stable control of contact has proven surprisingly difficult for robots. The subtlety of this problem is disguised by the ease with which humans manipulate. The fundamental cause of contact instability will be identified and experimentally-verified necessary and sufficient conditions for preserving stability on contact will be presented. The biological relevance of these results will be demonstrated by an experimental measurement of human arm behavior. This combination of theory and experiment show that intelligent control of physical systems requires a broader perspective than management of systems which deal only with information.

Introduction

There would seem to be a natural synergy between biology and robotics. Endowing a machine with the intelligence to replicate the skilled behavior of biological systems is a compelling and inspirational concept (to say nothing of its practical uses) and the exercise of building such a machine promises deep insights into the functioning of biological systems. But matching biological capability is a difficult and elusive goal. This paper will consider one example of an apparently simple capability of biological motor controllers which has so far proven surprisingly difficult to duplicate in robots. This example will be used to illustrate how a comparative study of robotics and biological movement control can yield deeper insights into both fields and to argue that intelligent control of mechanical systems requires a broader perspective than management of systems which deal with information alone.

One of the impediments to gaining a deeper understanding of biological motor control and a clearer perspective on the abilities to be expected of intelligent machines is that biological systems can be quite deceptive; superficial inspection (or introspection) rarely reveals the true complexity of what is going on. In biological systems, we see the solutions rather than the underlying problems. For example, it is easy to fall into trap of reasoning that "I walk with ease, therefore walking is easy". Yet successful walking machines are still largely a robotic researcher's dream. Despite concerted effort, machines that walk have only recently taken their first faltering steps outside the protected environment of the research laboratory, and though impressive progress has been made, no machine has yet demonstrated anything remotely close to the agility of biological systems.

Contacting the Real World

The situation in manipulation is no better, if not worse. Contact with objects is fundamental to manipulation, yet the apparently trivial problem of controlling the force exerted on a surface has proven surprisingly difficult. The "obvious" approach is to measure the force of contact and send that information to the controlling computer so that it may adjust or regulate the force exerted. Unfortunately, that approach is thwarted by a phenomenon which has been termed *contact instability*. Robotics researchers in numerous laboratories have reported that a robot which is capable of executing unrestrained motions stably and accurately will break into a pathologically uncontrollable chattering instability on contact with a rigid surface, bouncing off the surface and impacting it repeatedly. This problem is presently regarded as one of the prominent challenges of robotics [17].

Yet biological systems clearly have little difficulty contacting and manipulating objects. This provides one simple example of the synergy which can exist between biology and robotics. The manifest dexterity of humans provides an "existence proof" that this problem admits at least one excellent solution. But to understand the nature of that solution, it is helpful to look inside the system, and that is perhaps more easily accomplished in a robot.

Information Processing is Not Enough

Why is contact difficult? Understanding and solving this problem requires a revision of perspective from that commonly used to think about computers and control systems. Computers are information-processing machines. One of the most common and effective

ways to think about them is as devices which operate on signals; input information is processed to produce output information. The same signal-processing perspective dominates the techniques currently used to analyze and design control systems. But control systems must interact with the real world and signal-processing is a seriously incomplete description of what can happen in the physical world.

The flaw of the information-processing perspective is that operations on signals imply a one-way interaction; inputs produce outputs and not vice-versa. However, the essential mechanics of interactions between real objects involves an exchange of energy. Energy exchange is fundamentally a two-way interaction — an observation first made by Newton in his familiar third law: "For every action there is an equal and opposite reaction". This is a fundamental and critical difference.

The two-way dynamic interaction due to contact between a robot and an object in its environment means that the contacted object becomes, in effect, part of the collection of physical objects (i.e. the robot and anything it touches) which interact with the robot's controlling computer. The result is a change in the robot's dynamic behavior. Control system engineers have long known that a change in the behavior of a controlled system can have profound effects — in particular, it may destabilize the control system.

Ensuring robustness of a controller — preserving its stability and performance in the face of change — has been one of the central concerns of control systems research. Unfortunately, the usual assumption is that the changes are small in some well-defined sense, or occur outside the machine's normal operating regime. Yet the changes induced by contact can be dramatic: consider two people shaking hands. The magnitude of the contact force is not small by any measure — indeed, it is frequently larger than that encountered in typical object manipulation. The frequency content of the contact forces is approximately that of the normal operation of the hand. Thus the usual assumptions are hopelessly invalid in this situation.

An Alternative Approach: Physical Systems Theory

An alternative is to work from a physical systems perspective rather than an information-processing perspective. Though much remains to be done, considerable progress towards a unified theory of physical systems has been made in the past decades [1,18,20]. One fact which emerges is that the dynamic equations used to model physical systems are considerably more structured than those used to describe general information-processing operations. Knowledge of that structure can be used to advantage. Starting from a physical

systems perspective, recent theoretical work has identified a necessary and sufficient condition for preserving a manipulator's stability while contacting any passive¹ object of arbitrary dynamic complexity. That analysis will be reviewed briefly here. Details may be found in [2,3,4,12].

Consider a manipulator, A, and an object, B, which interact physically at a common point of contact. For simplicity we will initially consider only a single degree of freedom, though the arguments generalize to multiple degrees of freedom. Given a common point of contact, the velocity at that point is the same for both objects.

$$v_a = v_b$$

From Newton's third law we know that the force exerted on object B is equal and opposite to that exerted on manipulator A.

$$f_b = -f_a$$

Consequently, the power transmitted to object B is equal to that transmitted from object A.

$$f_b v_b = -f_a v_a$$

where the negative sign indicates that power is transferred out of system A. This is consistent with a general principle of physical systems theory that the interaction between two objects may not generate energy.

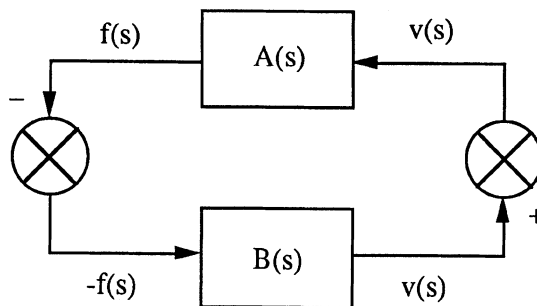


Figure 1: Block diagram representation of a system composed of two objects coupled at a common point of contact.

¹ In physical terms, an object is passive if the amount of energy which can be drawn from it is finite; it cannot supply power indefinitely.

If we describe the dynamic behavior of each object as an input/output relation between force and motion at the common point of contact, the interaction between the two objects may be represented in block-diagram form as shown in figure 1. Note that the interaction resembles a feedback connection of two signal processing operations in a computer or a control system. However, it is important to realize that unlike signal processing operations, the ways in which the two objects may interact physically are restricted. For example, a cascade connection of signal processing operations is common and useful, but a cascade connection of the input/output relations representing objects A and B would be physically meaningless [12].

To proceed further we will initially assume that for operation in the vicinity of the nominal conditions of contact the dynamic equations describing both objects may be linearized. We next assume that object B is passive and stable in isolation. The linearized input/output relation for a passive object $B(s)$ is a positive-real function [2,3] of s , where s is the Laplace variable. If we consider its steady-state response to a sinusoidal input, passivity requires that the average power transmitted to the object must be non-negative. Describing force and velocity at the point of contact as input and output, in steady state each will be a sinusoid. Considering the entire class of passive objects, the relative magnitudes of the two sinusoids are unrestricted but passivity requires that the relative phase angle between the two may not exceed $\pm 90^\circ$.

Because the interaction between the two objects resembles a feedback connection of two systems, we now may analyze the stability of the interaction between the manipulator $A(s)$ and the entire class of passive objects $B(s)$ using the Nyquist criterion [2,3]. The necessary and sufficient condition for stable feedback interaction between two systems $A(s)$ and $B(s)$ is that the Nyquist plot of the relation $A(s)B(s)$ must not encircle the -1 point on the complex plane. The Nyquist plot of the relation $A(s)B(s)$ may be obtained by rotating the Nyquist plot of the manipulator $A(s)$ through the angle determined by the Nyquist plot of the class of objects $B(s)$ — no more than $\pm 90^\circ$ — and multiplying it at each point by the magnitude of the Nyquist plot of the class of objects $B(s)$ — an arbitrary magnitude. From this it can be seen that the necessary and sufficient condition for stability of the coupled system is that the Nyquist plot of the manipulator $A(s)$ must lie exclusively in the right half of the complex plane. But that means that the linearized input/output relation of the manipulator must be a positive-real function, i.e. its input/output relation would be indistinguishable from that of a passive object.

The input/output relation describing the dynamic behavior of the manipulator at the point of contact is termed its driving-point impedance². It is important to realize that, unlike the usual stability conditions of conventional control theory, the condition derived above is a restriction on the manipulator's driving-point impedance rather than a restriction on the system eigenvalues. The eigenvalues of a system may be strictly stable without restricting its driving-point impedance to be positive-real. In fact, the linearized dynamic equations of a manipulator which is stable when unconstrained are guaranteed to have stable eigenvalues, yet experience has shown that that is not sufficient to guarantee stability on contact.

Note also that other aspects of the linearized dynamic behavior, for example the transfer function relating the desired position of the contact point to its actual position, need not be positive-real functions. The interesting point is that although the manipulator, with its controllable actuators, is intrinsically an active object, its stability will be preserved on contact if and only if it appears from an external viewpoint to exhibit a behavior equivalent to that of a passive physical object.

This result can be extended to the case of objects with nonlinear behavior. A brief sketch of the proof is provided here — details may be found in [2]. If the driving point impedance of the manipulator corresponds to that of a passive physical system, then the energy of that equivalent passive physical system may be used as a Lyapunov function for stability analysis. If the object B is described by nonlinear dynamic equations but it is nevertheless passive and stable when not in contact with the manipulator, then its energy may also be used as a Lyapunov function. To analyze the stability of the two systems, manipulator and object, when in contact we may use their total energy as a candidate Lyapunov function. Because each system is stable when not in contact with the other, the energy of each is a strictly non-increasing function of time. From physical systems theory it is known that the interaction between two objects may not generate energy. Therefore the total energy of the two systems is also a non-increasing function of time. Hence a manipulator with a driving-point impedance equivalent to that of a passive physical object will remain stable when in contact with a passive physical object of arbitrary dynamic complexity or nonlinearity.

The same line of reasoning can be used to extend the results to cases in which the manipulator is described by nonlinear dynamic equations, rather than by the linearized equations represented by $A(s)$. Once again, a brief sketch of the argument is provided here

² A generalized mechanical impedance is a dynamic operator which specifies the forces an object generates in response to imposed motions. It may be thought of as a dynamic generalization of the familiar notion of an elastic spring.

— details may be found in [12]. In the absence of a controller, a manipulator is a physical object with an associated energy function (e.g. the gravitational and kinetic energy of a typical robotic mechanism). Equilibrium posture of the manipulator is determined by the controller. If we assume that the steady-state behavior of the controller is such that it permits the definition of a potential function analogous to energy with at least a local minimum at the equilibrium posture, then the total "energy" of the manipulator plus the controller may be used as a Lyapunov function for stability analysis. To ensure stability of the isolated manipulator in the vicinity of the equilibrium posture it is sufficient that the controller define an equivalent "dissipation" function which guarantees that the total "energy" is a strictly non-increasing function of time. In effect, the behavior of such a controller is equivalent to that of a passive physical object.

The sufficient condition for isolated stability of the controlled manipulator is also sufficient to guarantee the stability of the manipulator when in contact with passive, nonlinear objects. The argument proceeds exactly as before: Because each system is stable when not in contact with the other, the energy of each is a strictly non-increasing function of time; interaction between two objects may not generate energy; therefore the total energy of the two systems is a non-increasing function of time. Thus a nonlinear controlled manipulator which emulates a passive, stable physical object will remain stable on contact with passive physical objects of arbitrary dynamic complexity or nonlinearity.

Note that this argument only yields a sufficient condition for coupled stability. However, the necessity of apparently passive behavior has been established by the local linearized analysis.

Experimental Verification

The validity of the theoretical result for computer-controlled mechanical systems has been confirmed experimentally [2]. A computer-controlled two-link robotic mechanism was used as a test platform. It was programmed with a series of different controllers and its performance and stability were measured while it was in contact with two classes of objects: unrestrained masses of various sizes and springs of various stiffnesses.

In one illuminating experiment the system was programmed to regulate its end-point position to follow a commanded trajectory in a small region in the center of its workspace. The controller used only position and velocity feedback information and was designed using the LQG/LTR technique, a widely-recommended modern approach to multi-variable controller design. However, that technique does not constrain the driving-point impedance of the mechanism to be that of a passive object.

In fact, both analytical prediction and direct measurement showed that the driving-point impedance imposed by the LQG/LTR controller violated the condition for contact stability. Consequently, the theory predicted that, on contact with certain classes of objects, the mechanism would exhibit instability. In fact, those objects included both sufficiently stiff springs and sufficiently large masses. Experimental observation confirmed these predictions. Although the controlled mechanism performed well when unconstrained, it was unstable on contact with a spring of stiffness 420 (± 20) N/m and on contact with a mass of 1.4 (± 0.1) kg.

It is important to note that in addition to confirming the theory, this experiment clearly demonstrated that contact instability is not merely a quirk of force-feedback controlled robots contacting rigid surfaces. It is a fundamental problem of contact. It may occur in the complete absence of force feedback and it may occur on contact with unrestrained objects.

In another experiment, the system was programmed with a simple impedance controller [12] designed to regulate the driving-point impedance of the mechanism. Measurement of the actual driving-point impedance of the system showed that it was indeed equivalent to that of a passive physical system, as dictated by the controller. Experimental observation also confirmed that this controller remained stable on contact with springs and masses which caused instability in other controllers, thereby confirming the theoretical predictions.

Biological Relevance

These experimental and theoretical results show that contact instability is a fundamental problem of manipulation. Do they provide any insight into how biological systems solve this problem? If this theoretical analysis is relevant, then the critical measurement to make is a determination of the driving-point impedance of the limbs. That information should provide important clues about the way the biological system controls manipulatory behavior.

The steady-state component of the mechanical impedance of a limb is a relation between displacements from a given posture and the restoring forces acting to maintain that posture. In general, the posture of a multi-joint limb is described by a vector, x , of positions (e.g. the location and orientation of the hand, the angles of the joints) and consequently the restoring forces, f , are described by a vector of equal dimension. Each of the components of the force vector may depend on any or all of the components of the displacement vector.

If the theory outlined above is relevant, then the behavior of the limb should be equivalent to that of a passive mechanical object. However, if the vector force-displacement relation is equivalent to that of a passive object, then it will be structured in a particular way [10,11]. The relevant passive physical object in this case is a generalized mechanical spring. From physical system theory it is known that the defining property of a mechanical spring is that it stores elastic potential energy. Consequently, the force-displacement relation for a generalized mechanical spring must be integrable so that the potential energy may be defined. Mathematically, this means that the *curl* of the vector relation between force and displacement should be zero if the steady-state behavior of the limb is to be equivalent to that of a passive physical object.

To test the passivity of the behavior, it is only necessary to consider small displacements. Although the force-displacement relation may be nonlinear for large displacements, for sufficiently small deviations about an equilibrium point it is approximately linear.

$$\mathbf{f} \approx -\mathbf{K} \, d\mathbf{x}$$

The stiffness $\mathbf{K}$, is a square matrix of coefficients. It can be partitioned into two components, one symmetric and one antisymmetric.

$$\mathbf{K}_s = (\mathbf{K} + \mathbf{K}^t)/2 \quad \mathbf{K}_a = (\mathbf{K} - \mathbf{K}^t)/2$$

The purely symmetric component, $\mathbf{K}_s$, represents the forces which have no curl and can be derived from a potential function. This is the part of the behavior which is equivalent to that of a mechanical spring. The antisymmetric component, $\mathbf{K}_a$, represents the forces which have non-zero curl and can not be derived from a potential function. Comparison of the relative magnitudes of the symmetric and antisymmetric components of the stiffness provides a numerical quantification of the extent to which the steady-state behavior of the neuromuscular system appears to be passive.

The mechanical impedance of a limb may be determined experimentally by displacing it from a given posture and measuring the resulting restoring forces. A series of psycho-physical experiments were performed on human subjects [14] to determine the steady-state force-displacement relation at the hand. Seated subjects grasped the handle of a computer-controlled manipulandum (similar to that described above) and held it underneath one of a planar array of target lights. The wrist was braced and the elbow was supported comfortably so that the forearm and upper arm moved in the horizontal plane intersecting the shoulder. Under computer control the handle of the manipulandum was displaced a distance of 4 or 8 mm in one of eight directions spaced equally at 45-degree intervals around a circle. Subjects

were instructed to count mentally to three and then move the handle in the direction opposite to the perturbation. This experimental design was chosen for two reasons: First, the cognitive task delayed the onset of the subject's voluntary response so that the limb was able to reach steady state following the perturbation before the onset of the voluntary response. Secondly, the subject's voluntary response could be clearly distinguished from the underlying involuntary response.

The motions of the handle and the forces exerted on the handle during were recorded continuously during each trial. When the limb reached steady state following a perturbation (determined by monitoring the magnitude of the velocity) the displacements and corresponding restoring forces were recorded. The observations were fit to a linear force displacement relation using multivariable regression techniques.

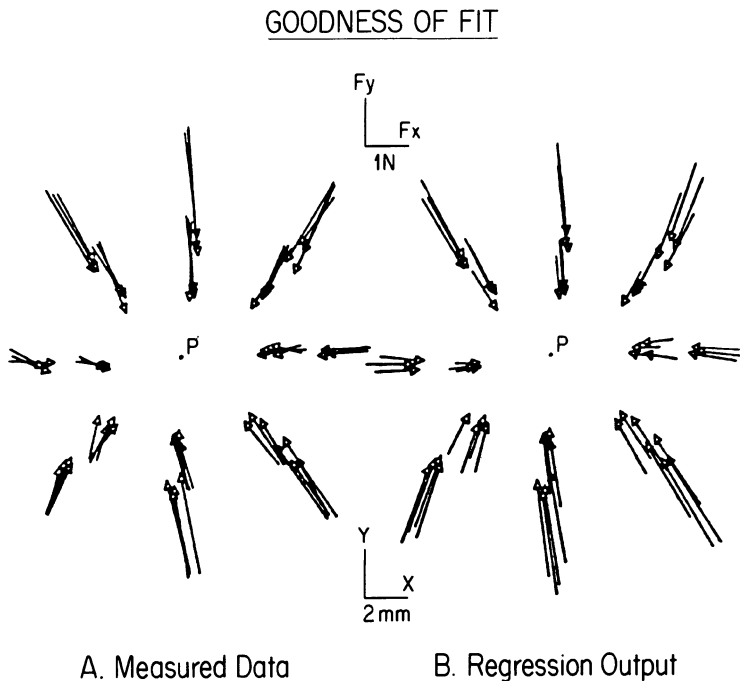


Figure 2. A vector diagram of the the restoring forces measured from a subject maintaining the hand at the equilibrium position P. Measured data is shown in a, forces reconstructed from the linear regression are shown in b.

Figure 2a shows a sample of the raw data. The force vectors are drawn as arrows; the distance from their heads to the equilibrium point P in the centre of the plot represents the

corresponding displacement. Figure 2b shows the force vectors generated by the best fit linear stiffness matrix for the same displacements and demonstrates that the linear model provides an excellent fit to the data.

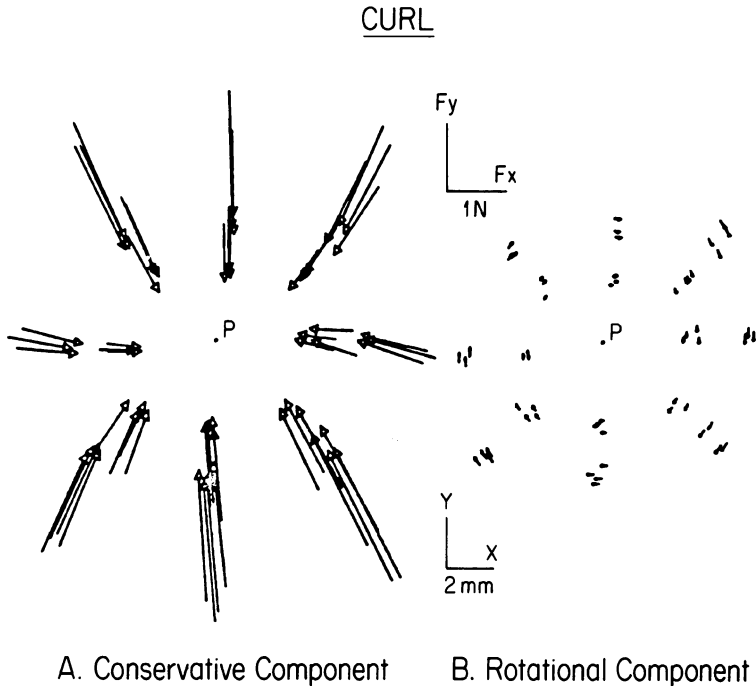


Figure 3. The spring-like or conservative component of the restoring forces are shown in a. The rotational or curl component is shown in b.

The stiffness matrix was decomposed as described above into symmetric and antisymmetric components. A typical sample of the force field corresponding to each of these components is shown in figure 3. The conservative component dwarfs the rotational component. In fact, in about half of the measurements, the curl was not statistically different from zero; in all cases, it was small. Clearly, the steady-state behavior of the neuro-muscular system is essentially indistinguishable from that of a mechanical spring, at least under the conditions of these experiments.

Implications for Neural Feedback

How surprising are these results? Could they have been predicted from available knowledge of muscle mechanics and neural control circuitry? Over a wide range of physiological conditions, the steady-state behavior of a single muscle has been shown to resemble that of a mechanical spring [5,6,19]. For a given level of alpha-motoneuron activity, the force generated by the muscle depends on its length. The force-displacement relation may be nonlinear, but it is integrable (every length corresponds to a unique force, though the converse may not be true). In the absence of feedback, the concatenation of an arbitrary number of muscles could not produce a non-spring-like behavior.

Can neural feedback produce non-spring-like behavior? Neural feedback relating the length of a muscle to the activity of the alpha-motoneuron which drives the same muscle acts to modify the relation between the force and length of a muscle, but the result is still equivalent to a mechanical spring [6,13,15,16]. If the descending commands to the neuro-muscular systems are fixed, the force it generates is an integrable function of its length. Therefore the concatenation of an arbitrary number of muscles with *inter*-muscular feedback could not produce a non-spring-like behavior.

However, neural feedback pathways which relate motions at one joint to the activity of muscles at another are known to exist. If this *intra*-articular feedback is not bi-laterally symmetric — for example, if the feedback-induced moment about the elbow per unit shoulder motion were not exactly equal to the moment about the shoulder per unit elbow motion — the vector force-displacement relation for the limb would exhibit a non-zero curl. In this case the steady-state driving-point impedance would not be equivalent to that of a passive mechanical spring. Thus there is no a-priori reason why the neuro-muscular system should be constrained to behave like a mechanical spring.

Consequently, the experimental observation that the curl is effectively zero implies that either the *intra*-articular feedback is not active under the steady-state conditions of the experimental measurements — and given the pathways exist, this would be puzzling — or the *neural feedback system is exquisitely balanced*. Whatever the explanation, the neuro-muscular system is clearly an active object — over the time-scales relevant for normal motor behavior, muscles can supply substantial amounts of energy. Nevertheless, the measured steady-state impedance of the arm is that of a passive object. This is consistent with the theoretical necessary and sufficient conditions for stability on contact with passive physical objects of arbitrary dynamic complexity. Given that intra-articular feedback pathways exist, it may

even be the case that the biological system goes to considerable pains to ensure that this condition is satisfied.

Conclusion

The purpose of this paper is to illustrate how a comparative study of robotics and biological movement control can yield deeper insights into both fields. By any reasonable definition, manipulation of an object requires physical contact with it, and maintaining stable behavior on contact with a wide range of physical objects is one of the fundamental prerequisites for dexterous manipulation. One of the benefits of a comparative study of robots and biological systems is a clearer definition of the problems the biological system has solved — experience with robots has shown that meeting this prerequisite is far from a trivial matter.

Conversely, a careful study of biological systems may reveal clues to the solution of these problems. In stark contrast to robotic actuators, the forces generated by biological actuators — muscles — are not solely a function of the input commands from the information-processing parts of the system (e.g. the central nervous system). Instead, muscle force depends also on other mechanical variables such as the length of the muscle, its velocity of shortening, etc. At first glance, this might appear to be a non-ideal complexity of the biological system, but the analysis reviewed above shows that this feature may be precisely that which guarantees stability on contact with a wide range of physical objects.

A comparative study of manipulation in robots and biological systems makes it clear that understanding intelligent control of mechanical systems requires a broader perspective than management of systems which deal with information alone. This conviction of the need for a new perspective has been the motivation guiding the author's work on impedance control [7,8,9]. Impedance control is an attempt to formulate an alternative approach to controlling mechanical systems. The unifying idea underlying impedance control is the concept of physical equivalence. This (unproven) conjecture states that for every physical system with a controller, there exists at least one physical system without a controller which exhibits the same behavior. If true, this means that the same analytical tools may be used for both aspects of an intelligent mechanical system; the informational transactions which characterize control system software can be described in the same context as the energetic transactions which characterize physical systems. The analysis reviewed herein is an example of the benefits of this approach.

Acknowledgements

Neville Hogan is supported in part by National Science Foundation Research Grant No. ECS-8307461, by National Institutes of Health Research Grants No. AR 40029 and NS 09343 and Office of Naval Research Grant No. N00014-88-K-0372.

References

1. Breedveld, P. C. (1984) Physical Systems Theory in Terms of Bond Graphs. Ph.D. Thesis, Technical University of Twente, Enschede, Netherlands.
2. Colgate, J. E. (1988) The Control of Dynamically Interacting Systems. Ph.D. Thesis, Department of Mechanical Engineering, M.I.T.
3. Colgate, J. E. and Hogan, N. (1988) Robust Control of Dynamically Interacting Systems. *Int. J. Control*, Vol. 48, pp. 65-88.
4. Fasse, E. D. (1987) Stability Robustness of Impedance Controlled Manipulators Coupled to Passive Environments. S.M. Thesis, Department of Mechanical Engineering, M.I.T.
5. Gordon, A. M., A. F. Huxley and F. J. Julian (1966) The Variation in Isometric Tension with Sarcomere Length in Vertebrate Muscle Fibers. *J. Physiol.*, 184: 170-192.
6. Hoffer, J. A. and S. Andreassen (1981) Regulation of Soleus Muscle Stiffness in Premammillary Cats: Intrinsic and Reflex Components. *J. Neurophysiol.*, 45: 267-285.
7. Hogan, N. (1980) Mechanical Impedance Control in Assistive Devices and Manipulators. paper TA-10B in Proceedings of the 1980 Joint Automatic Control Conference, San Francisco.
8. Hogan, N. (1984) Impedance Control of Industrial Robots. *Robotics and Computer Integrated Manufacturing*, Vol. 1, pp. 97-113, 1984.
9. Hogan, N. (1985) Impedance Control: An Approach to Manipulation: Part I - Theory, Part II - Implementation, Part III - Applications. *ASME Journal of Dynamic Systems, Measurement, and Control*, pp. 1-24.
10. Hogan, N. (1985) The Mechanics of Multi-Joint Posture and Movement Control, *Biol. Cybern.*, 52: 315-331.
11. Hogan, N. (1986) Multivariable Mechanics of the Neuromuscular System. Proceedings of the 8th Ann. Conf. of the IEEE/Eng. in Med. and Biol. Soc. pp. 594-598.
12. Hogan, N. (1988) On the Stability of Manipulators Performing Contact Tasks. *IEEE Journal of Robotics and Automation*, Vol. 4, pp. 677-686.
13. Matthews, P. B. C. (1959) "The Dependence of Tension Upon Extension in the Stretch Reflex of the Soleus Muscle of the Decerebrate Cat", *J. Physiol.*, 147: 521-546.
14. Mussa-Ivaldi, F. A., Hogan, N. and Bizzi, E. (1985) Neural and Geometric Factors Subservicing Arm Posture. *J. Neuroscience*, Vol. 5, pp. 2732-2743.
15. Nichols, T. R. and J. C. Houk (1973) Reflex Compensation for Variations in the Mechanical Properties of Muscle. *Science*, 181: 182-184.
16. Nichols, T. R. and J. C. Houk (1976) Improvement in Linearity and Regulation of Stiffness that Results from Actions of Stretch Reflex. *J. Neurophysiol.*, 39: 119-142.

17. Paul, R. P. (1987) Problems and Research Issues Associated with the Hybrid Control of Force and Displacement. Proceedings of the IEEE Conference on Robotics and Automation, pp. 1966-1971.
18. Paynter, H. M. (1961) Analysis and Design of Engineering Systems. M.I.T. Press, Cambridge.
19. Rack, P. M. H. and D. R. Westbury (1969) "The Effects of Length and Stimulus Rate on Tension in the Isometric Cat Soleus Muscle", J. Physiol., 204: 443-460.
20. Rosenberg, R. C. and D. C. Karnopp (1983) Introduction to Physical System Dynamics, McGraw Hill, New York.

MOTOR CONTROL SIMULATION OF TIME OPTIMAL FAST MOVEMENT IN MAN

Blake Hannaford* and Lawrence Stark

University of California Berkeley

*Now at Department of Electrical Engineering FT-10

University of Washington

Seattle, WA 98195

In the fastest human movements, three pulses of activation can be recorded from the agonist and antagonist muscles in the form of bursts of the electromyogram, (EMG). Our detailed simulation of the muscles and load enabled quantitative determination of the Action, Braking, and Clamping roles of the first agonist, antagonist, and second agonist bursts respectively. These results are obtained by varying EMG based control signal inputs to a fixed sixth order non-linear model of horizontal head rotation. In an additional simulation experiment it is shown that a three pulse control signal is required for optimal movement when the movement is constrained to a small target window beginning soon after movement start. Similarly, experimental variations in time optimal movements obtained by application of external loads, or alteration of proprioception by tendon vibration can be accurately simulated with the same unchanged model but with changed experimental conditions and an input control signal which is changed by the same amount as observed EMG changes.

The Electromyogram and Control Pulses: A common feature of the electromyogram (EMG) in very fast human movements around a single degree of freedom is the tri-phasic activation pattern^{1,2,3,4}. This characteristic pattern of activation consists of three bursts appearing in the agonist, antagonist and agonist muscles, respectively. Optimal control theory has contributed predictions of the control signal required to change a system from one state to another in minimum time. Specifically, the optimal control theorists of the sixties^{5,6,7} showed that with limited actuator force available and an idealized inertial load, a two pulse signal is sufficient for time optimal movement. Clark and Stark⁸ applied the maximum principle of Pontryagin to the eye movement system and determined optimal two pulse and three pulse control signals. To some extent, the roles of the initial pulses of activation are intuitive. For example, it is clear that the first agonist pulse (PA) is responsible for initiating the movement and that the antagonist burst (PB) stops the movement, appearing when the movement is performed too fast for braking by passive forces⁹. The

role of the second agonist burst PC has been less clear. Ghez and Martin⁴ for example, have speculated that it may have a viscous damping function, dissipating any excess energy remaining in the system after attainment of final target position. However, typically movement velocity is near zero during PC. Three activation bursts have also been found by simulations using an inverse model driven by recorded movement dynamics¹⁰

Protocol for Simulation Studies: The result of the studies reported here is a more precise definition of the role of each of the three components of the classical tri-phasic EMG signal. This is achieved through simulating the dynamics of the muscles and plant driven by control signals derived from the timing of recorded EMG activity. The horizontal head rotation system is simulated in this paper by a sixth order non-linear model described by a set of six state equations. These equations were integrated by digital computer to compute the simulation.^{11,12}

Three simulations were performed with the model: In the Pulse Ablation technique, we used a rectangular control signal having fixed timing and variable amplitudes, obtained values for the three pulse amplitudes, and explored the effects of removing control signal pulses on resulting dynamics. The approach taken here was minimization of the mean squared error between model output and experimental data. When this RMS error is minimized, the best fit possible under the assumed control signal model is obtained.

Enumeration of the control signal space is a long computation in which the space generated by the parameters of the control signal model is exhaustively searched for points resulting in movements meeting some validity criterion. This approach was used by Lehman and Stark¹³ in simulating 65536 (2^{16}) eye movements in approximately 7 seconds on the ILLIAC IV.

In the Control Signal Variation technique, a model input signal based on EMG recordings was altered in accordance with EMG pulse area changes quantitatively observed during altered experimental conditions such as external loads. The simulated dynamical changes can then be compared with those observed experimentally.

Pulse roles clarified by ablations: When the complete three-pulse control signal was used as model input, the output model dynamics closely matched an experimentally recorded 40deg movement (Fig. 1i trace ABC). The role of each pulse in the movement was revealed by simulations in which pulses were selectively removed (ablated) from the control signal (Fig. 1i, traces AB and A). Trace AB is the model output resulting from a control signal consisting of only pulses A and B. The AB trace was identical with the ABC trace until 215 ms after the onset of PA. The resultant position trace attained the desired final position, but then drifted away from target position. This negative drift was also evident on examination of the velocity trace in which the negative velocity phase had been extended in duration in the absence of PC. Trace A is the result of a simulation in

which only the PA pulse was supplied. The resulting trajectory was a movement of 64° , approximately 150% of the intended (target) amplitude. The peak velocity attained by this movement (609°sec^{-1}) was the same as that of the complete (ABC) movement, but the positive velocity phase was extended in the absence of PB, and the velocity remained positive throughout the extent of the movement. This corresponded to the lack of a negative phase of the acceleration curve in the absence of PB.

These results suggested that the three control signal pulses can be labeled A, B, C to suggest the functions **Action**, the initial pulse that starts the movement, **Braking**, the antagonist pulse that initiates deceleration, and **Clamping**, the final agonist pulse that fixes final position.

Pulse Ablations with Constant Movement Amplitude: A similar set of simulations were computed in which the movement amplitude was held constant (Fig. 1ii). In these simulations, the pulse heights were recomputed to preserve the correct final state (defined as position = final target position, and velocity = 0.0). Trace ABC is the same three-pulse simulation as in Fig. 1i. It represents the best three-pulse fit to experimentally measured time-optimal movements. As the control signal contains fewer pulses, movement time increases.

Additional constraints on the control signal were necessary in some circumstances to generate all the control signals by the optimization process. For example, in order to compute the optimum three-pulse fit, the ABC control signal of Fig. 1i and 1ii, an additional constraint was necessary, namely that pulse C be greater than a certain minimum value. Without this constraint, the optimization process found that a two-pulse signal gave a better fit in spite of the EMG evidence for three control signal pulses. This problem was solved through enumeration experiments by searching for a constraint under which PC was necessary for time optimality. In the enumeration simulations, software control loops were placed around an existing, verified simulation program to iterate selected control signal parameter values through 10 values between 0 and 50 kg. Each resulting control signal was simulated by the model and the fastest resulting movement subject to the validity criterion was selected. Each enumeration took approximately 1 week of computation time on an MC68000 based computer. The validity criterion consisted of a small window around the target which the movement may not leave, beginning a short time (t_{end}) after the start of simulation.

Five enumeration simulations were run in which the end times were 220 ms, 235 ms, 240 ms, 280 ms, and 350 ms. The 220 and 235 ms runs produced no valid movements (the control signal ended at 250 ms). When the endtime was 350 ms, a two pulse control signal was optimal. It is immediately apparent (Fig. 1iii & 1iv) that as the end condition window is brought closer to the beginning of the movement, the optimal control signal

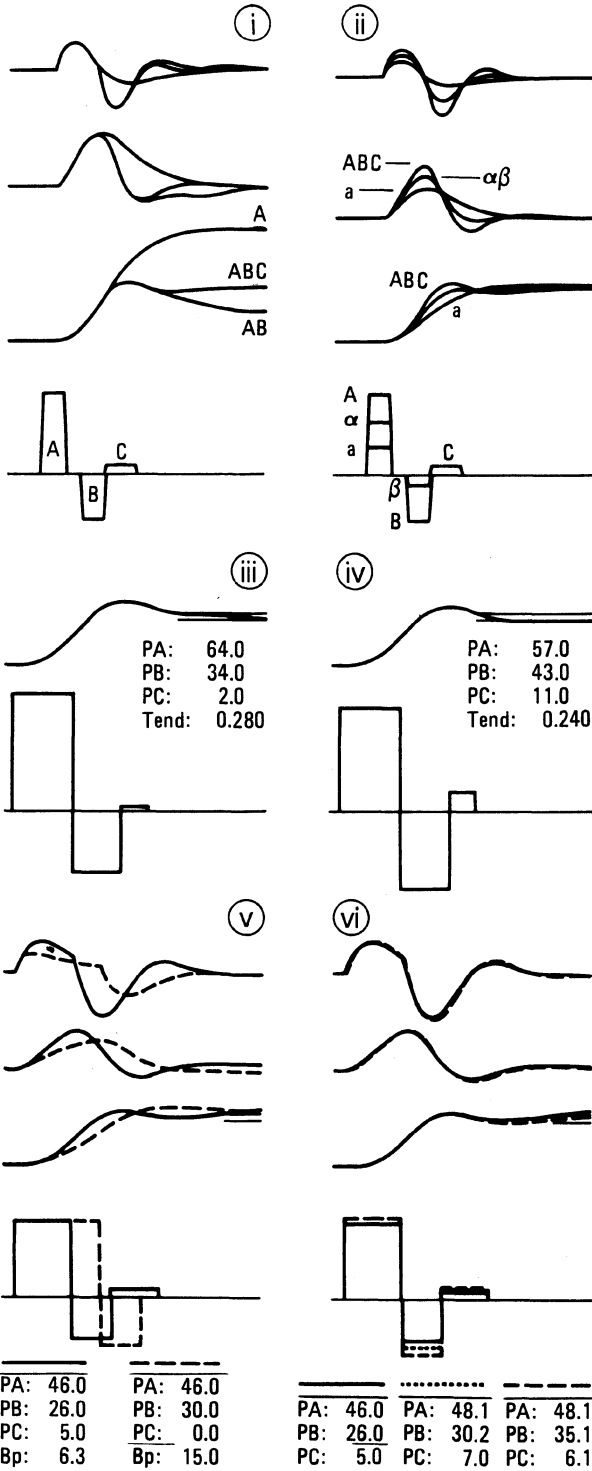


FIGURE LEGEND

Fig. 1. Simulation studies of fastest possible (time optimal) voluntary human movements which are characterized by three pulses of muscle activation manifested in Electromyogram (EMG). All simulations are of time optimal, voluntary, 40 degree human horizontal head movement. (i) Illustration of the pulse ablation technique in which dynamic effects of three activation pulses are revealed by removal of one or more control signal pulses. Trace ABC, complete simulation which is optimal fit (mean square sense) to experimentally recorded data; AB, simulation driven by first two pulses; A, effect of initial agonist pulse alone. (ii), simulations identical to (i) but amplitudes have been re-computed to preserve final position. Trace ABC, same as in (i); $\alpha\beta$, optimal two pulse signal; trace "a", optimal single pulse signal. Note addition of control signal pulses reduces time of arrival at target. (iii)(iv) Results of enumeration study in which all possible three pulse control signals were simulated and fastest resulting movement meeting validity criterion (two horizontal lines near target) was selected. In enumerations in which validity criterion began soon after beginning of movement (iv) a significant third pulse is required to generate the optimal movement. (v) Simulation of effects of viscous loading on time optimal movement. When control signal is altered according to EMG changes observed after subjects adapt to additional viscous load (dashed line) and model is altered to reflect the added load, dynamical changes match those observed experimentally. (vi) Control signal variation based on EMG changes observed in tendon vibration experiments and resulting dynamics (short dashes; agonist vibration, long dashes: antagonist vibration). Traces identified as follows: panels (i), (ii), (v), (vi), from top to bottom, acceleration, velocity, position, agonist control signal, antagonist control signal. Panels (iii) and (iv), top to bottom, position, agonist control signal, antagonist control signal.

contains a significant clamping pulse. Notice that the earlier the start of the acceptance window, the less overshoot is observed in the simulated movement.

This enumeration study showed that PC could be generated by explicit specification of an optimality criterion (peak velocity) and a movement acceptance criterion (the simple end window). When the end window is sufficiently close to the end of the control signal, then a substantial PC is required in order to generate the fastest movement to the target. This correspondence with the tri-phasic EMG pattern suggests that the subject designs a control signal which demands quick convergence along with maximum velocity. However, the additional activity (PC) is not generated by an actual position error because the constraint begins after the control signal ends. Thus the clamping pulse is evoked by the end condition.

Control Signal Variation - Added Viscosity: In one set of experiments¹⁴ EMGs and head rotation dynamics were recorded during time optimal movements. In addition to the normal movements, a viscous load was attached to the head, and the subjects attempted more fast movements under the altered conditions. The movements selected for analysis from both the normal and loaded experiments were the fastest of those made after practice.

With a continuous added viscous load, subjects adapted their control strategies to make movements about 75 - 80% as fast as their unloaded time-optimal movements. The resulting EMG envelopes showed a reduction in PC amplitude. Compensation for the viscous load caused PA to increase in width and PC to decrease in amplitude. Peak velocity and acceleration were reduced. A heuristic simulation in which PA was increased in width, PB in height, and PC reduced to zero was performed (Fig. 1v, dashed lines). For comparison, a normal movement was also simulated (solid lines). The added viscous load was modeled by an increase of 150% in the plant viscosity parameter, Bp. The resulting simulated trajectory is slower than the normal 40 degrees time optimal movement, and is more damped, reaching its maximum value about 100 ms later. The resulting dynamics agree well with the experimental data¹⁵

Control Signal Variation - Continuous Vibration: As with the addition of viscous load, the effects of continuous vibration were analyzed after the subject had the chance to adapt to the altered conditions¹⁶ The results depended on whether the vibration was applied to the agonist or antagonist muscle tendon. Simulations were performed (Fig. 1vi) in which the height of each control signal pulse was changed according to the EMG pulse-area changes measured by planimetry from the experimental data. The decrease in peak velocity observed with agonist continuous vibration (short dashed lines) is not observed in the simulation. With antagonist continuous vibration (long dashed lines), the simulation of Figure 1vi shows a decrease in final position. This is consistent with the experiments of Roll and Vedel¹⁷ who found undershoot in presence of antagonist vibration.

Discussion: This report represents an overview and synthesis of experimental and simulation approaches to the study of time optimal 1 degree-of-freedom human movements. By altering the model control signal we have gained insight into the non-causal, non-linear control laws used.

Simulating the effects of removing or ablating the different control signal pulses revealed the role of each pulse in shaping the dynamics of the movements and yielded the mnemonic designations: Action, Braking, and Clamping for the first agonist pulse, the first antagonist pulse, and the second agonist pulse, respectively. These roles have recently been confirmed experimentally. Wierzbicka, et al.¹⁸ stimulated biceps and triceps in vivo with pulses of current based on EMG recorded during fast movement. Adjustment of the amplitude of the current pulses produced a matching movement in the stimulated arm. The pulse ablation experiment was then performed, with identical results to our simulation (Figure 1i).

The enumeration studies address the interaction of optimality and accuracy of movement. The results suggest that a third burst of EMG activity is required for time optimal movement when position constraints are applied immediately after the end of the movement control signal. Finally, when external loads or signals are applied which alter the conditions under which the movement is performed, resulting changes can be measured in EMG envelopes. When we changed model control signals by the same amount, we observed similar dynamical changes to those observed in the experiments. The sixth order non-linear model we have used thus accurately predicts the relationship between EMG activity and movement dynamics for changes about a reference movement.

References

1. K. Wachholder and H. Altenburger, "Beitrage zur Physiologie der willkurlichen Bewegung," *Pflugers Arch. Ges. Physiol.*, vol. 214, pp. 642-661, 1926.
2. R.W. Angel, "Antagonist muscle activity during rapid arm movements: central vs. proprioceptive influences," *Journal of Neurology, Neurosurgery, and Psychiatry*, no. 40, pp. 683-686, 1977.
3. M. Hallett and C.D. Marsden, "Physiology and Pathophysiology of the ballistic movement pattern," in *Motor Unit types, Recruitment & P. in H. & D., Prog. clin. Neurophys.*, ed. J.E. Desmedt, vol. 9, pp. 331-346, Karger, Basel, 1981.
4. C. Ghez and J.H., Martin, "Control of Rapid Limb Movement in the Cat. III. Agonist-Antagonist Coupling," *Exp. Brain Research*, vol. 45, pp. 115-125, 1982.
5. R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, 1965.
6. R. E. Kalman, "When Is a Linear Control System Optimal?," *Trans. ASME J. bas. Engineering*, vol. 86D, pp. 51-60, 1964.

7. L. S. Pontryagin, V. C. Boltyanskii, R. V. Camkrendze, and E. F. Mischenko, *The mathematical theory of optimal processes*, Interscience Publishers, John Wiley & Sons, New York, 1962.
8. M. Clark and L. Stark, "Time Optimal Behavior of Human Saccadic Eye Movements," *IEEE Trans. Automatic Control*, vol. AC-20, pp. 345-348, 1975.
9. F. Lestienne, "Effects of Inertial Load and Velocity on the Braking Process of Voluntary Limb Movements," *Experimental Brain Research*, vol. 35, pp. 407-418, 1979.
10. B. Hannaford, W.S. Kim, S.H. Lee, and L. Stark, "Neurological Control of Head Movements: Inverse Modeling and Electromyographic Evidence," *Math. BioScience*, vol. 78, pp. 159-178, 1986.
11. W.H. Zangemeister, S. Lehman, and L. Stark, "Sensitivity Analysis and Optimization for A Head Movement Model," *Biol.Cybern.*, vol. 41, pp. 33-45, 1981b.
12. B. Hannaford and L. Stark, "Late Agonist Burst (PC) Required for Optimal Head Movement: a Simulation Study," *Biological Cybernetics*, vol. 57, pp. 321-330, 1987.
13. S. Lehman and L. Stark, "Simulation of Linear and Nonlinear Eye Movement Models: Sensitivity Analysis and Enumeration Studies of Time Optimal Control," *J. Cyber. & Inf. Sci.*, vol. 4, pp. 21-43, 1979.
14. B. Hannaford, M.H. Nam, V. Lakshminarayanan, and L. Stark, "Electromyographic Evidence of Neurological Controller Signal with Viscous Load," *Journal of Motor Behavior*, vol. 16, pp. 255-274, 1984.
15. B. Hannaford, *Control of Fast Movement: Human Head Rotation*, Ph.D. Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1985.
16. B. Hannaford, G. Cheron, and L. Stark, "The Effects of Applied Vibration on the Tri-Phasic EMG Pattern in Neurologically Ballistic Head Movements," *Experimental Neurology*, vol. 88, pp. 447-460, 1985.
17. J.P. Roll and J.P. Vedel, "Kinaesthetic Role of Muscle Afferents in Man, Studied by Tendon Vibration & Microneurography," *Exp. Brain Research*, vol. 47, pp. 177-190, 1982.
18. M.M. Wierzbicka, A.W. Wiegner, and B.T. Shahani, "Role of Agonist and Antagonist Muscles in Fast Arm Movement in Man," *Experimental Brain Research*, vol. 63, pp. 331-340, 1986.

Constraints on underspecified target trajectories*

Michael I. Jordan
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

*This project was supported in part by BRSR 2 S07 RR07047-23 awarded by the Biomedical Research Support Grant Program, Division of Research Resources, National Institutes of Health and by a grant from Siemens Corporation. The results presented in this paper appeared previously in Jordan (1990).

Much of the recent interest in artificial neural networks is founded on the development of supervised learning algorithms for nonlinear problems [1, 30, 39, 42, 47]. These algorithms, the most well-known being backpropagation, are able to model a large class of nonlinear transformations by assigning credit to internal “hidden” units. The remaining units—those connected directly to the environment—are generally assumed to be provided with target states. This assumption appears to be a liability; it is by no means clear that such desired outputs can always be provided. Consider, for example, a network serving as a feedforward controller for a robot. Such a network must produce torques as a function of the environmental goal and the current state of the robot. In general, however, the environment provides only the goal and not the torques that achieve the goal. Furthermore, if we assume the existence of an oracle that provides the torques as training data, then there appears to be little reason (other than perhaps speed) not to use the oracle as the controller in place of the network.

Recently, Jordan and Rumelhart [22] have proposed a solution to this problem (see also [37, 48]). Their approach extends the applicability of the supervised learning paradigm to a general class of problems in which desired output vectors are specified implicitly. The idea is essentially to treat the output units of a network as hidden units in a larger network that incorporates a learned *forward model* of the environment.

The forward modeling approach has interesting consequences for the interpretation of activation patterns in a network. Generally speaking, units in a network are either “visible” or “hidden” (although units may change their roles over trials [40]). Visible units are directly connected to the environment; their activations are explicitly determined by the task and have a direct environmental interpretation. The activations of hidden units are generally underdetermined by the task and have no direct a priori interpretation. These distinctive properties of visible units and hidden units are conflated in a network that incorporates a forward model. In particular, the output units (viz., the *articulatory* units, see below) of such a network are hidden with respect to learning and yet are visible given their direct connection to the environment. The former fact implies that the activations of these units need not be uniquely determined by the task; this allows us to deal with indeterminate learning problems. The indeterminacy can be resolved by making use of additional constraints on the activations. Furthermore, the fact that the output units have a direct environmental interpretation implies that additional constraints on these units also have a direct environmental interpretation. Indeed, the interpretation may provide a basis for choosing the form of the constraints. Ideally, such constraints should apply broadly across tasks and reflect generally desirable properties of performance in a given domain. In the current paper, I present examples of such constraints in the domain of motor learning and discuss the implementation of constraints as components of a network learning rule.

Terminology

The following terminology is used throughout the paper. *Task space* is the vector space in which environmental specifications of tasks are given to the learner (see [44] for a related use of the term). For motor learning problems, task space vectors are movement

outcomes such as positions in space, geometric relations between objects, acoustic spectra, or elapsed time. *Articulatory space* is the vector space of possible control signals to the effectors. In control systems terminology, articulatory vectors represent the input to the plant, whereas task space vectors represent the output from the plant (including possible kinematic transformations). I use the term “environment” as a synonym for “plant.”

Forward models of the environment

The approach pursued here treats motor learning as the optimization of a quantity—a cost functional—defined in task space, subject to various additional constraints. The major theoretical problem in this framework is that of ensuring that all of the terms needed for optimization are in fact available to the learner. There are several cases in which this assumption seems problematic. In particular, when the controller must interact with an unknown external system or when the biophysical plant undergoes change due to maturation or injury, then the solutions derived by optimization change accordingly. Thus if motor learning is to be characterized as a process of optimization, then the underlying learning algorithm must itself be reconfigurable and sensitive to a changing environment. In the current framework, this problem is addressed via the notion of a *forward model* of the environment ([22]). In terms of its input/output behavior, a forward model is closely related to notions of “internal simulation,” “efference copy,” or “predictive model” that have been considered by many authors (e.g., [7], [13], [33]): A forward model is a learned internal mapping that allows the system to predict the results it expects to obtain, given the state of the environment and the output of the controller.¹ Somewhat surprisingly, a forward model also allows the system to go backwards; that is, to find a controller output that achieves a desired result. Consider, for example, a simple positioning problem in which the system must achieve a task-space target $\mathbf{z}^*$. We define a cost functional that is the weighted sum of squares of the difference between a desired result $\mathbf{z}^*$ and an actual result $\mathbf{z}$:

$$J = \frac{1}{2}(\mathbf{z}^* - \mathbf{z})^T \mathbf{H}(\mathbf{z}^* - \mathbf{z}), \quad (1)$$

where $\mathbf{z}^*$ and $\mathbf{z}$ are vectors in task space and $\mathbf{H}$ is a weighting matrix. The vector $\mathbf{z}$ is a function of the environmental state $\mathbf{s}$, and the controller output $\mathbf{x}$:

$$\mathbf{z} = \mathbf{f}(\mathbf{s}, \mathbf{x}). \quad (2)$$

The controller output $\mathbf{x}$ is in turn a function of the internal state $\mathbf{q}$, an input vector $\mathbf{p}$ (the “plan”), and a set of structural parameters $\mathbf{w}$ that describe the current configuration of the controller:

$$\mathbf{x} = \mathbf{g}(\mathbf{q}, \mathbf{p}, \mathbf{w}). \quad (3)$$

¹The forward modeling approach has applications to learning both feedforward control and feedback control. In this paper, I discuss only the former; thus, “controller” means “feedforward controller” unless otherwise stipulated.

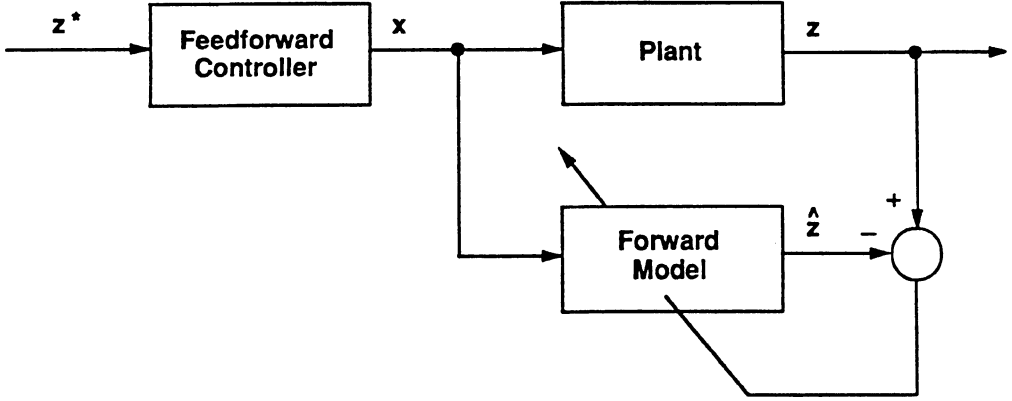


Figure 1: Learning the forward model.

The problem is to find a set of parameters $\mathbf{w}$ so that J is minimized; that is, so that the desired result $\mathbf{z}^*$ and the actual result $\mathbf{z}$ are equal. To solve this optimization problem, we first find the gradient of J with respect to the parameters $\mathbf{w}$:

$$\nabla_{\mathbf{w}} J = -\frac{\partial \mathbf{x}}{\partial \mathbf{w}}^T \frac{\partial \mathbf{z}}{\partial \mathbf{x}}^T \mathbf{H}(\mathbf{z}^* - \mathbf{z}). \quad (4)$$

For this gradient to be useful in a learning rule, each of its terms must be available to the learner, either separately or in combination. The vector $\mathbf{z}^* - \mathbf{z}$ is the task-space error and can assumed to be provided by the environment. The matrix $\frac{\partial \mathbf{x}}{\partial \mathbf{w}}$ describes a credit assignment process internal to the controller. The Jacobian matrix $\frac{\partial \mathbf{z}}{\partial \mathbf{x}}$ is the term whose origin appears to be problematic. It is the matrix that relates small changes in the controller output to small changes in the task space results, and cannot be assumed to be available a priori, or provided by the environment. However, all of the derivatives in the matrix are *forward* derivatives. They are easily obtained by differentiation if a forward model is available. To accomodate the possibility of an unknown or changing environment, the forward model must itself be learned, but this can be achieved directly by system identification. Once the model is accurate over a particular domain, its derivatives provide a learning operator that allows the system to convert errors in task space into errors in articulatory space, and thereby change the controller.

The structure assumed by the forward modeling approach is shown in block-diagram form in Figure 1 and Figure 2 and in network form in Figure 3. The network is a concatenation of two subnetworks, namely, the controller subnetwork and the forward model subnetwork. The controller subnetwork connects the *plan* units and the *state* units to the *articulatory* units. The plan units are the input units of the network; they receive activation from higher levels and designate the particular control surface to be instantiated. The state units receive recurrent connections from within the network and have a dual role: They

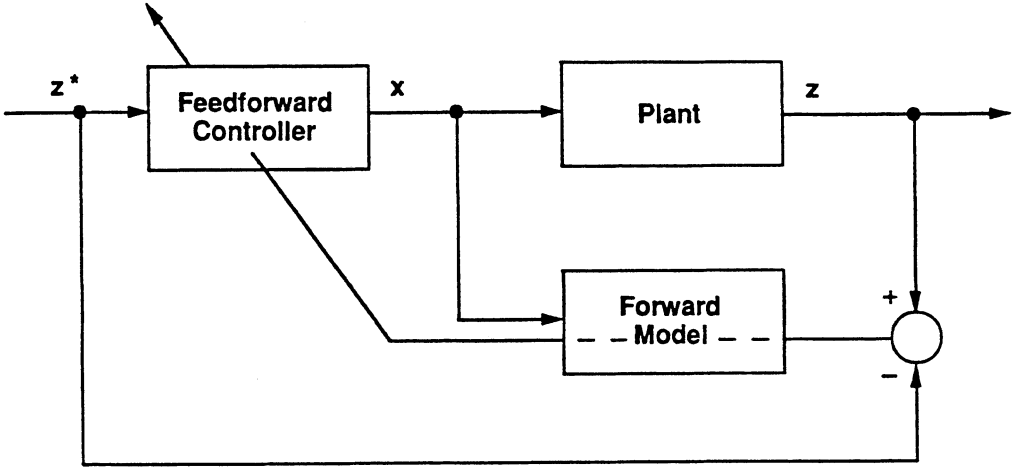


Figure 2: Using the forward model to learn an inverse model.

provide internal state for the system, thereby allowing the network to autonomously generate sequences of actions [20], and they also estimate the state of the environment, thereby giving the network the capability to control the environmental dynamics. The controller subnetwork also includes intermediate “hidden” units, so that a nonlinear function may be learned [42]. The articulatory units are the output units of the network; they are connected to the effectors.

The forward model subnetwork takes its input from the articulatory units and from those state units that encode the environmental state (the connections implementing the latter dependency are not shown in the figure). The output of the forward model is a pattern of activation $\hat{z}$ on the *task* units; it is the network’s internal estimate of the task space vector z .

Learning the forward model

Figure 1 shows the processing needed for learning the forward model. This learning process is a straightforward application of supervised learning techniques. Given a particular state s of the plant and a particular action x , the resulting task space vector z is used as a target for the output of the forward model. The weights of the forward model are changed to reduce the following error measure:

$$E = \frac{1}{2}(z - \hat{z})^T(z - \hat{z}),$$

where $\hat{z}$ is the output of the forward model. This error measure goes to zero as the network becomes a more accurate predictor of task-space results.

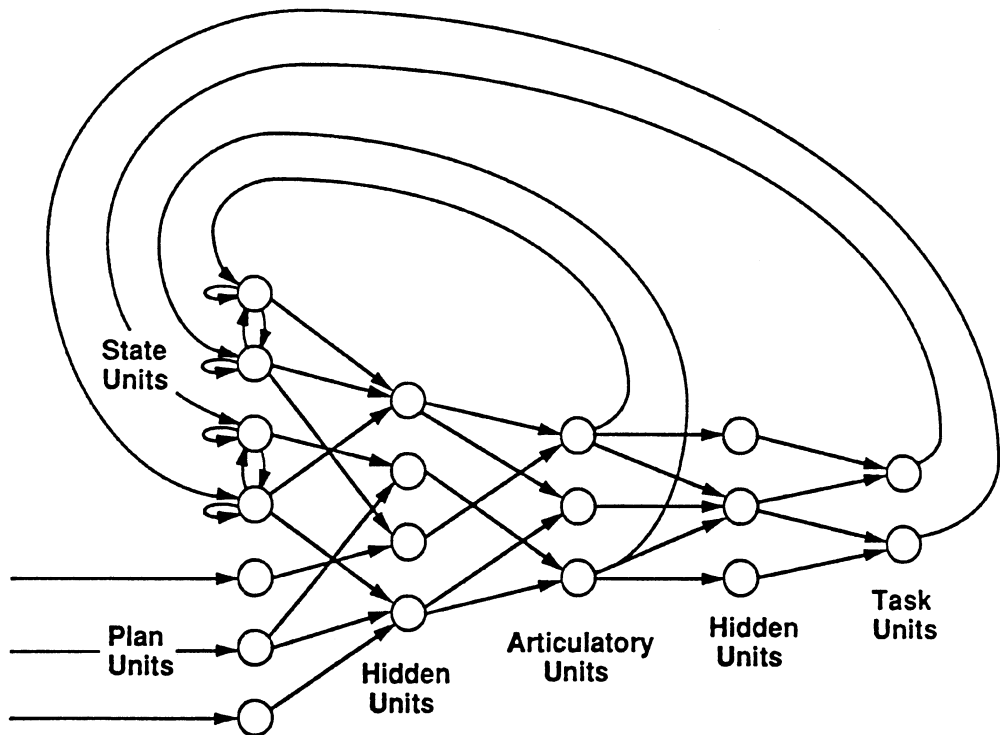


Figure 3: The general network architecture. The articulatory units are the output units of the network.

Learning the controller

Once the forward model is sufficiently accurate over a particular domain, it can be used in learning the feedforward controller in accordance with Equation 4. The required computation is depicted in Figure 2, in which a dotted line passing backward through a transformation represents multiplication by the transpose Jacobian matrix of the transformation.

Let us assume for simplicity that a target trajectory $\mathbf{z}^*(t)$, $0 < t < T$, is provided to the network. Then at each time step the weighted error vector $\mathbf{H}(\mathbf{z}^*(t) - \mathbf{z}(t))$ is transformed by two successive transpose Jacobian matrices according to Equation 4. In the network, this computation is achieved by using the backpropagation algorithm to propagate the weighted error vector backwards through the forward model and down into the controller [22]. The propagated vector need not depend on the model output $\hat{\mathbf{z}}$, although activation must pass forward in the forward model subnetwork so as to instantiate the correct transpose Jacobian matrix (which varies as a function of the articulatory configuration) for the backward pass.

It is important to note that the gradients computed by propagating a vector backward through a forward model are estimates, and are exact only if the forward model is exact (over a particular domain). Exact gradients, however, are not necessary for the system to find an exact solution to the learning problem [18]. This is a consequence of the use of the actual task space vector $\mathbf{z}$ rather than the estimated task space vector $\hat{\mathbf{z}}$ in the error term in Equation 4. A corollary of this fact is that the forward model and the controller can be learned concurrently, with a concomitant speedup in learning time, if some mechanism (e.g., a feedback controller) is available to guide the system into the appropriate domain (see [3, 24] for related use of a feedback controller). If no such mechanism is available, then the forward model must be learned separately in a phase of random exploration before the controller can be learned.

For many problems, it is not necessarily the case that a target trajectory is available. For example, a target may be available only at the final time step T . In order to assign credit in this case, it is necessary to propagate the final configuration error vector backward “in time,” either by using an algorithm that saves past values of activation [42, 49] or computes ongoing traces of activations [50]. These algorithms are directly applicable to the network in Figure 3; the backward propagation process can in effect continue backwards through the recurrent connections at the state units and repeatedly back through the forward model subnetwork and the controller subnetwork in order to compute the necessary gradients.

Indeterminacies and intrinsic constraints

In many motor learning problems, the task does not uniquely determine the control actions that must be produced. For example, when the mapping from articulatory space to task space (Equation 2) is many-to-one, or when targets are provided intermittently in time, there are in general an infinity of articulatory solutions possible at each time step. In such cases, the forward modeling approach behaves gracefully; the articulatory error vector simply points in the direction of the manifold of articulatory solutions. This is a major virtue of the forward modeling approach—indeterminate problems can be presented as such

to the network. By avoiding computing explicit target vectors for the articulatory units, the approach allows additional constraints on the articulatory solution to be incorporated directly into the learning algorithm. Such constraints can be chosen to reflect generally desirable properties of action such as smoothness, distinctiveness, and speed. In this section I present a brief discussion of such constraints and their implementation in the learning algorithm.

Smoothness constraints allow for a reduction in the amount of information needed to specify a trajectory [15]. Also, in systems with excess degrees of freedom, smoothness in articulatory space implies that task-space goals nearby in time are achieved with similar articulatory configurations [18]. This allows a system to move more rapidly through a sequence of task-space goals than would otherwise be possible.

A general form of smoothness constraint can be written as a condition on the weighted mean squared velocity of the articulatory vector $\mathbf{x}$:

$$J_s = \frac{1}{2} \sum_{i=2}^N (\mathbf{x}_i - \mathbf{x}_{i-1})^T \mathbf{K} (\mathbf{x}_i - \mathbf{x}_{i-1}), \quad (5)$$

where the subscript i denotes the time step. If $\mathbf{x}$ includes components that are themselves velocities or accelerations of underlying variables, then J_s may be interpreted in terms of mean square acceleration or jerk.

When actions have a communicatory purpose, it is sensible that they be distinct so that they may be easily interpreted [31, 32, 34]. A *distinctiveness constraint* can be obtained as the negative of a smoothness term, enclosed in an exponential term for stability:

$$J_d = \frac{1}{2} \sum_{i=2}^N e^{-(\mathbf{x}_i - \mathbf{x}_{i-1})^T \mathbf{L} (\mathbf{x}_i - \mathbf{x}_{i-1})}. \quad (6)$$

It is also useful to incorporate constraints associated with deviations of the articulatory configuration from a rest configuration $\mathbf{x}^{(0)}$ [32, 36, 45]. Such a constraint has an interpretation in terms of energy expenditure or loss of predictability for large deviations. A *rest-configuration constraint* is given as the weighted sum of squares of these deviations:

$$J_r = \frac{1}{2} \sum_{i=1}^N (\mathbf{x}_i - \mathbf{x}^{(0)})^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}^{(0)}). \quad (7)$$

Many authors have proposed that action can be characterized in terms of the optimization of quantities such as energy, time, smoothness, distinctiveness, and information (e.g., Hogan, 1984; Lindblom, 1983; Lindblom, 1986; Mussa-Ivaldi, McIntyre, & Bizzi, 1988; Nelson, 1983). What distinguishes the current approach is that such principles are viewed not only as descriptions of the endstate of skill, but also as active constraints on the process of learning. As is demonstrated in the sequel, it is of particular interest to study the effects of these constraints when they apply simultaneously in different coordinate systems.

Network implementation of intrinsic constraints

The full specification of the active constraints on the learning of a particular task is given as a composite cost functional. This cost functional is a sum of task constraints and intrinsic constraints, weighted by a set of regularization parameters $\{\gamma_i\}$. The intrinsic constraints on the articulatory output involve additional error terms computed at appropriate locations in the network in Figure 3. The weights are changed according to the sum of all of the propagated error terms. Thus, for example, an articulatory smoothness constraint is implemented with additional error terms obtained by computing temporal differences at the articulatory units,² as can be seen by differentiating Equation 5:

$$\nabla_{\mathbf{w}} J_s = - \sum_{i=2}^N \left\{ \frac{\partial \mathbf{x}_{i-1}}{\partial \mathbf{w}}^T \mathbf{K}(\mathbf{x}_i - \mathbf{x}_{i-1}) + \frac{\partial \mathbf{x}_i}{\partial \mathbf{w}}^T \mathbf{K}(\mathbf{x}_{i-1} - \mathbf{x}_i) \right\}. \quad (8)$$

Note that the first term in this equation depends on partial derivatives at the $i - 1^{\text{st}}$ time step and causes $\mathbf{x}_{i-1}$ to approach $\mathbf{x}_i$. Thus, this term leads to anticipatory behavior. The second term causes $\mathbf{x}_i$ to approach $\mathbf{x}_{i-1}$ and therefore yields perseveratory behavior.

It is similarly straightforward to compute gradients for distinctiveness constraints and rest-configuration constraints. (Distinctiveness gradients are equivalent to smoothness gradients where each term in the temporal difference vector is multiplied by the scalar $z_i = -e^{-(\mathbf{x}_i - \mathbf{x}_{i-1})^T \mathbf{L}(\mathbf{x}_i - \mathbf{x}_{i-1})}$). Several such constraints can be incorporated in the learning rule. For example, a network may attempt to produce outputs that have distinct perceptual effects but are nonetheless smooth in articulatory space. Such a network is described by the following cost functional:

$$J = \frac{1}{2} \left\{ \sum_{i=1}^N (\mathbf{z}_i^* - \mathbf{z}_i)^T \mathbf{H}_i (\mathbf{z}_i^* - \mathbf{z}_i) + \gamma_1 \sum_{i=2}^N e^{-(\mathbf{z}_i - \mathbf{z}_{i-1})^T \mathbf{K}(\mathbf{z}_i - \mathbf{z}_{i-1})} + \right. \\ \left. \gamma_2 \sum_{i=2}^N (\mathbf{x}_i - \mathbf{x}_{i-1})^T \mathbf{L}(\mathbf{x}_i - \mathbf{x}_{i-1}) + \gamma_3 \sum_{i=1}^N (\mathbf{x}_i - \mathbf{x}^{(0)})^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}^{(0)}) \right\},$$

which generates two additional error terms at the task units and three additional error terms at the articulatory units.

Simulations

This section presents the results of simulation experiments that illustrate some of the properties of the forward modeling approach. Examples are given of problems in kinematics, dynamics, the use of feedback, and the learning of input parameters. In all examples, the emphasis is on the problem of indeterminacy.

²In the case of smoothness constraints and distinctiveness constraints, it is necessary for units to save their activation at the preceding time step so that the appropriate temporal differences can be formed.

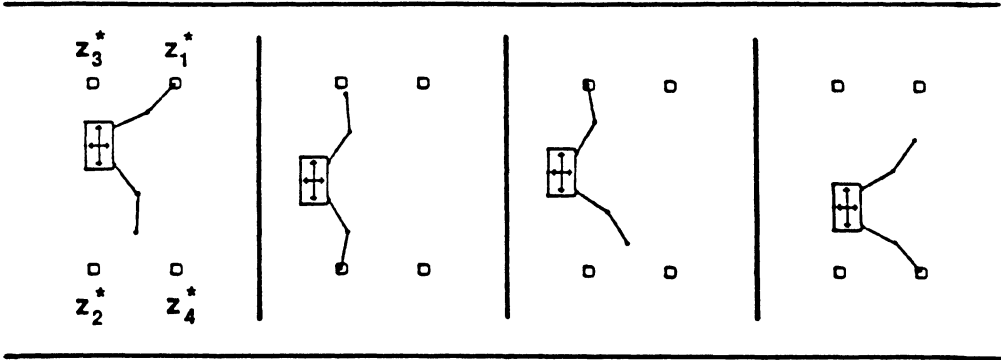


Figure 4: A six-degree-of-freedom manipulator learning a sequential positioning task. The system returns to the initial frame from the final frame.

Smoothness in a multi-effector system

The first example is an inverse kinematic learning problem. The manipulator shown in Figure 4 is a two-fingered "hand" with two translational degrees of freedom and four revolute joints, yielding a 6-dimensional articulatory space. The task the system must learn is a sequential positioning task in which either one of the endpoints is required to touch a target at each time step. Target positions are specified in Cartesian coordinates, therefore task space is 2-dimensional. There is also an intermediate 4-dimensional coordinate system that encodes the positions of the endpoints of the fingers. The mapping from endpoint space to task space utilizes an additional binary variable to specify the "active" finger (the finger that is assigned to the current target). The assignment of fingers to targets is made dynamically during learning, based on a distance comparison.

The network used to control this manipulator is of the general form shown in Figure 3. The network has six articulatory units for the manipulator's six articulatory degrees of freedom and an additional articulatory unit to encode the active finger. The activations of the first two articulatory units are interpreted as translational displacements and the next four articulatory units represent the sines of the angles of the four joints. These units are connected to a set of four endpoint units through an intermediate layer of six hidden units. The endpoint units and the active-finger unit are connected to two task units through a layer of four hidden units. The network has six plan units, and four hidden units in the motor program subnetwork. There are recurrent connections from the endpoint units to a set of four state units so that serially ordered behavior can be produced.

In a simulation, learning of the forward model subnetwork was carried out in a separate exploratory phase. This was done by repeatedly choosing random activation patterns $\mathbf{x}$ for the articulatory units (chosen from a uniform distribution), and learning the forward kinematic mapping from articulatory space to endpoint space. It is not necessary to learn

the mapping from endpoint space to task space because that mapping is independent of the environment. Furthermore, appropriate values for the weights in the path from the endpoint units and active-finger unit to the task units can be determined a priori. After approximately 2,000 exploration trials, the weights in the forward model subnetwork were held fixed, while a set of cyclic sequences was learned.

During the sequence-learning phase, targets were presented sequentially to the task units, while a smoothness constraint was imposed on the articulatory units. The results for a particular sequence are shown in Figure 4. As can be seen, the network finds a solution to the problem that meets the task constraints. Furthermore, due to the smoothness constraint, the network makes use of the excess degrees of freedom to anticipate and persevere other actions in the sequence. Each configuration involves a nonlinear compensatory relation between the translational degrees of freedom and the joint angles of the active finger; furthermore, the joint angles of the inactive finger show parallel anticipation of the following configuration.

The results of this simulation are suggestive of phenomena of coarticulation in speech production ([26, 32, 38]). In speech production, the articulatory configuration associated with the production of a given phoneme changes according to its temporal context. Not all aspects of the configuration are equally malleable, however. For example, in the production of an [s], there are strong constraints on the position of the tongue tip in the vicinity of the alveolar ridge, so that fricative output in a certain frequency band is achieved. There are weaker constraints on other articulators, such as the tongue dorsum and the lips. In the production of the velar fricative [x], there are strong constraints on the position of the tongue dorsum and weaker constraints on the tongue tip and the lips. In the current framework, all such constraints are assumed to be implicit in the specification of a task constraint in acoustic terms, in conjunction with a learned forward model of the nonlinear mapping from articulation to acoustics. The smoothness constraint leads to varying amounts of anticipation by each of the articulators, within the implicit bounds established by the inverse images of the acoustic constraints.

Another point worth noting concerns the stability of the learned sequence. Simulations show that the solution found by the network is stable; in particular, it is a periodic attractor both in articulatory coordinates and in task coordinates. When the system is perturbed by adding transient noise to the activations of the articulatory units or the task units, it returns to the learned cyclic sequence, demonstrating the stability of the solution. Furthermore, the same results are obtained for the other three sequences learned by the network. Thus, several stable trajectories can be learned together in the network.

Smoothness in the endpoint space of a multi-effector system

Consider the case of a musician playing a lengthy ascending scale on the piano. In learning such scales, students are taught to cross the thumb under the index finger and the middle finger, so that every third note is played by the thumb.

In an analog of the piano-playing task, the network used in the previous experiment learned a linear sequence in which the finger assignments were alternated (the assignments

were imposed from outside the network). Simulations revealed that the network readily learned the task, but was unable to find solutions in which one of the fingers crossed over the other. Rather, the solutions exhibited a small tendency toward crossing over and a larger tendency to use the translational degrees of freedom. Changing the gains in favor of the joint angles simply abolished whatever anticipatory behavior was present. The reason for these results is clear—in articulatory space, a posture with the fingers crossed is very different from an uncrossed posture. Thus, a smoothness constraint in articulatory space is inconsistent with a transition between these types of posture. Why then do piano players adopt the solution of crossing the thumb under?

Reconsideration of the task reveals that it is not smoothness in articulatory space that is relevant to the scale-playing task, but smoothness in *endpoint* space. The closer the endpoint of an unused finger is to the next target in sequence, the more quickly the transition between targets can be made. That is, given the nature of the task, the potential speed of transitions in endpoint space is more important than their cost in articulatory terms.

For the network described in the previous section, endpoint space and task space are the same. Smoothness in task space can potentially yield contextual effects in this example because there are four degrees of freedom in task space versus the two degrees of freedom needed to specify a target. Figure 5 shows the results of a simulation in which a smoothness constraint was applied to the task units, rather than to the articulatory units. As can be seen, the network finds a solution in which the endpoint positions are anticipated even though articulatory positions are not anticipated.

This simulation demonstrates that smoothness in different coordinate systems may lead to qualitatively different kinds of behavior. Part of the learning process may be the discovery of the appropriate coordinate systems in which to apply such constraints.

Distinctiveness in task space; smoothness in articulatory space

Rapid communication would seem to require articulatory representations to be smooth in time, whereas the resulting perceptual representations should be distinct. In a system with excess degrees of freedom, these two tendencies need not be incompatible with one another. This possibility was investigated in a simulation using a distinctiveness constraint in task space and a smoothness constraint in articulatory space.

The manipulator used for this experiment is the same as in the previous two experiments, although task space is defined somewhat differently. For the distinctiveness constraint to apply properly, task space is defined as the two-dimensional space representing the endpoint position of the active finger. In the network, a seventh articulatory unit designates the active finger, and the mapping from this unit and the four endpoint units to a pair of task units constitutes an additional layer in the forward model subnetwork.

Targets for the task are rectangular regions in task space (this is achieved by using inequality constraints on the task units). The results are shown in Figure 6. As can be seen, due to the distinctiveness constraint, the endpoints tend to move toward the outer corners of the rectangles defined by the task constraints. Nevertheless, the effects of the

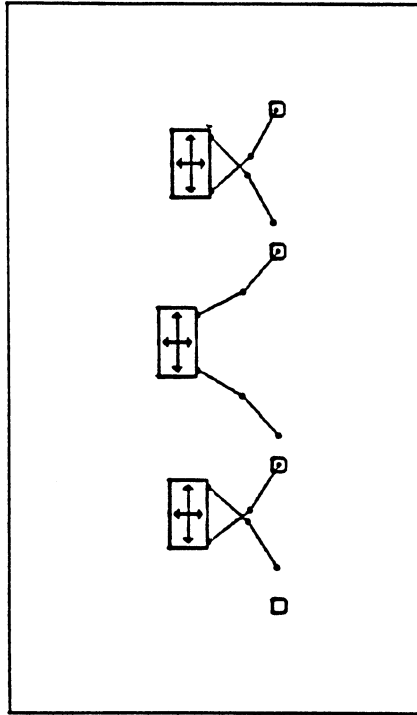


Figure 5: Learning a sequential positioning task with a smoothness constraint on the task units. Three successive time steps are overlaid in this figure.

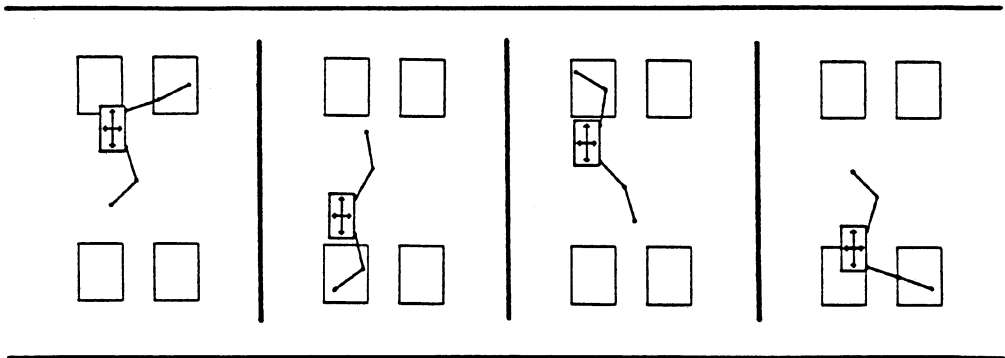


Figure 6: The combined effects of a distinctiveness constraint and a smoothness constraint.

smoothness constraint on the excess degrees of freedom are still in evidence; in particular, the position of the unconstrained endpoint tends to anticipate its position later in the sequence.

Feedback and feedforward

In control system terminology, the network used in the previous simulations is a feedforward controller with state (i.e., internal dynamics). I have proposed that the feedforward controller is learned by processes that estimate the parameters of a forward model of the controlled plant, and then use those parameters in forming gradients of a cost functional defined in terms of the output of the plant.

Traditionally, a feedforward controller is augmented with an error-correcting feedback controller [2]. The feedback controller corrects for unanticipated disturbances and provides control in those regions of state space for which the feedforward signal has yet to be learned. An example of such a composite control system for a dynamic arm is given in the following section. The current section provides an example of a composite feedforward-feedback system for a kinematic plant. The focus here is on the way in which excess degrees of freedom are utilized over the course of learning. To this end, I consider a network that can perform a sequence correctly, although slowly and awkwardly, on the first trial.

A feedback controller forms an error vector based on the difference between a reference vector and the output of the plant. This error vector is then transformed into a control vector. Note that this process is very similar to the forward modeling learning rule developed earlier. In fact, a feedback controller for a kinematic plant can be obtained by taking the gradient of the cost functional in Equation 1 with respect to the control vector $\mathbf{x}$ rather than with respect to the weights $\mathbf{w}$. The resulting control law is the locally linear law given by:

$$\Delta \mathbf{x} = -\mu \frac{\partial \mathbf{z}}{\partial \mathbf{x}}^T \mathbf{H}(\mathbf{z}^* - \mathbf{z}), \quad (9)$$

where μ is a gain. As before, the transformation due to the transpose Jacobian matrix can be computed using a learned forward model. The initial value of $\mathbf{x}$ is provided by the feedforward controller. The equation describes an error-correcting scheme that iteratively updates the control vector $\mathbf{x}$ so as to null the performance error $\mathbf{z}^* - \mathbf{z}$. (An essentially identical control law has been proposed in a model of frog limb movements [5]. It is also of interest to consider a closely related control law in which the left-hand side is in units of generalized force, rather than displacement. Such a law can be used for a dynamic plant, due to the static relationship holding between the transpose Jacobian matrix $\frac{\partial \mathbf{z}}{\partial \mathbf{x}}^T$ and the generalized forces in the domain and range of the corresponding function [16, 28, 43]).

The feedback controller makes it possible for the system to perform a sequence correctly on the first trial. Consider an overall system structure containing an outer loop and an inner loop. The outer loop initializes the target $\mathbf{z}^*$ to the current vector in the sequence being learned and passes it to the feedforward controller. The output of the feedforward controller is a vector $\mathbf{x}$ that provides the initial control signal for the inner loop. The inner loop simply applies Equation 9 (the feedback controller) repeatedly until the magnitude of

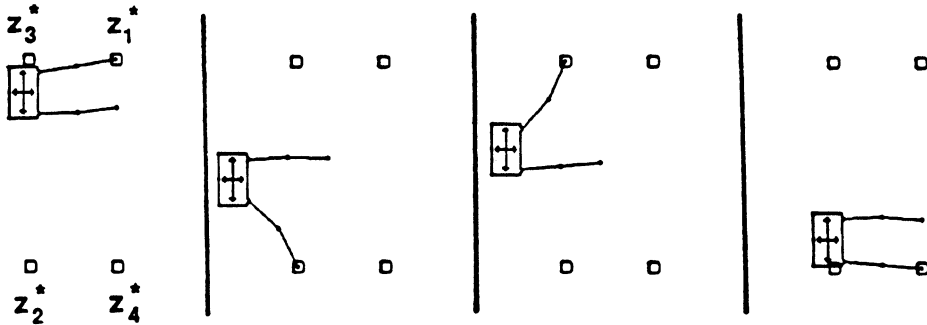


Figure 7: Performance of a feedback controller. The figure shows the configurations found by the feedback controller for each target in the sequence.

the error vector $\mathbf{z}^* - \mathbf{z}$ is less than some criterion value. At this point, the outer loop resets the target to the next vector in the sequence.

As the feedforward controller is changed through learning, the initial control vectors that it provides are more nearly correct, so that fewer iterations through the feedback controller are needed to reach the targets. This is demonstrated by the simulation results shown in Figure 7 and Figure 8, which display the output of the system on the last iteration of the inner loop for each target. Figure 7 shows the results before any learning of the feedforward controller has taken place, whereas Figure 8 displays the results after twenty learning trials. A total of 42 iterations of feedback were required to perform the sequence in Figure 7, whereas only 22 iterations were needed in Figure 8. Of course, once the feedforward control has been learned completely, as in the simulation shown in Figure 4, no feedback iterations are required.

Learning of the feedforward controller leads not only to improvements in the speed of performance, but also to increasingly effective use of excess degrees of freedom. As shown in Figures 7, 8, and 4, the system shows an increasing ability to use excess degrees of freedom in anticipating future actions. This occurs even though the system is at a stage in which performance on the task is essentially correct with respect to the task constraints. Such optimization has often been noted in qualitative descriptions of motor learning [9, 12].

Trajectory formation for a dynamical arm

Problems of indeterminacy are not restricted to issues of kinematic redundancy but also arise when trajectories must be generated. For example, a physical arm has mass and therefore moves along a spatially continuous trajectory. If a task specifies only a final position for the endpoint of the arm, then an infinite number of solutions—that is, trajectories that arrive at the target—are possible. In this section, I discuss how this indeterminacy problem can be resolved within the same computational framework as that demonstrated for the kinematic

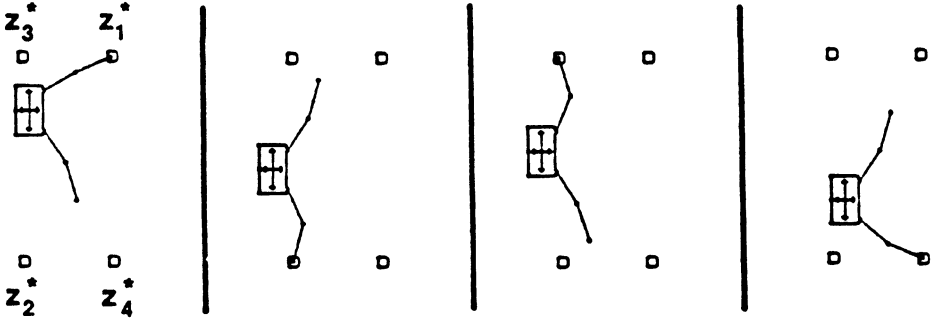


Figure 8: Performance of the composite system, after partial learning of the feedforward controller.

case in previous sections.³

Due to the elastic properties of muscles, the control of arm movement can be characterized in terms of a (virtual) equilibrium position to which the arm would move if efferent activity were held constant [15]. Evidence from arm movements in monkeys suggests that this virtual equilibrium position shifts gradually over the course of a movement toward the final position [6]; thus, control can be treated in terms of a “virtual trajectory” [15]. If the muscles are modeled as second-order linear systems, then the torque τ is generated as a function of the virtual trajectory $u(t)$ according to:

$$\tau = -\beta\dot{\theta} - k(\theta - u), \quad (10)$$

where θ is the actual position of the arm, β is a viscosity parameter, and k is a stiffness parameter. For the simulations presented in this section, the control law in this equation is assumed, and the task is to learn a virtual trajectory $u(t)$, given a specification of a desired final position.

In [22] the problem of learning the inverse dynamics of a two-joint arm is used to illustrate how the forward modeling approach applies in the case of environments with state. In such a case, the forward model is a learned mapping from the current state and control signal (positions, velocities, and torques) to the next state (accelerations, integrated twice). Backpropagation through the forward dynamic model allows the controller subnetwork to find torques that cause the arm to follow desired trajectories.

The current problem differs from that of inverse dynamics in that we are not assuming that a desired trajectory is available. When the only signal available for learning is a final-position error, it is necessary to propagate the error backwards “in time” so that the entire control trajectory can be changed. This can be accomplished in the current framework by identifying the state units of the forward model subnetwork with the state

³A closely related approach to trajectory formation has been proposed by Kawato (1989).

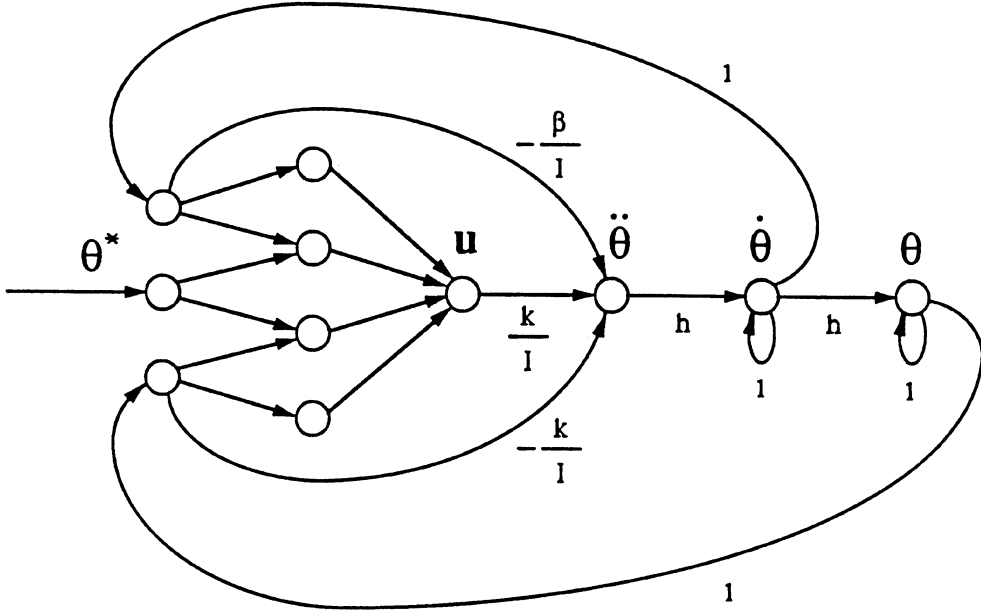


Figure 9: A network that learns virtual trajectories for a one-joint arm.

units of the controller subnetwork and propagating errors through the resulting recurrent network. An example of such a network is shown in Figure 9 for the case of a one-joint arm. In the diagram, the state units have been drawn to the right, to more easily depict their interconnection structure. The network is nonetheless a special case of the general network shown in Figure 3. In this network, the forward model is implemented by weighted links arriving at a single task unit representing the joint acceleration. In the figure, I have assumed that the weights on these links have already been learned. As can be seen, the output of the forward model is therefore given by:

$$\ddot{\theta} = -\frac{\beta}{I}\dot{\theta} - \frac{k}{I}(\theta - u),$$

which is the forward dynamics of the one-joint arm.

Backpropagation of the final-position error in this network can be augmented with a smoothness constraint to determine a unique virtual trajectory. Smoothness in task space corresponds to the minimum jerk principle suggested in [15]. In this case, the functional minimized by the network is the following:

$$J = \frac{1}{2}[\theta^* - \theta(N)]^2 + \gamma \sum_{i=1}^N [\ddot{\theta}(i) - \ddot{\theta}(i-1)]^2, \quad (11)$$

where θ^* is the desired final position.

Simulations reveal that the network is able to find virtual trajectories that minimize jerk based only on the final-position error. The time needed to learn such trajectories can be lengthy, however. A network learning four trajectories required approximately 19,000

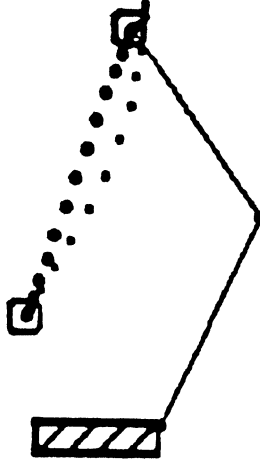


Figure 10: Virtual and actual trajectories for a two-joint arm.

iterations. One approach to decreasing this learning time is to use the desired final position as a control signal, as was originally proposed in the mass-spring model of arm movement [8, 25, 41]. The virtual trajectory is a sum of this constant signal and the time-varying signal being learned:

$$u(t) = \theta^* + u'(t),$$

where $u'(t)$ is the output of the articulatory unit.⁴ Simulations show that this approach leads to a learning time nearly an order of magnitude faster than before (1,950 iterations).

In principle, it is straightforward to generalize the network in Figure 9 to the case of a multijoint arm. Multiple units are required for vector variables and a hidden layer is required in the forward model to represent the nonlinear interaction terms. A forward kinematic model is also required so that position errors can be computed in Cartesian coordinates. Learning based on the final position error then proceeds as before. Figure 10 shows the results of a simulation of a network that learned four trajectories for a two-joint arm. The heavy dotted line is the trajectory followed by the endpoint of the arm (a minimum jerk constraint was used), whereas the light dotted line is the virtual trajectory discovered by the learning algorithm. The simulation used feedforward of the desired final position and required 6,670 iterations.

In the multi-joint case, new considerations arise concerning the choice of a smoothness

⁴Note that final-position control is mathematically equivalent to a proportional-derivative feedback controller; thus, this approach is conceptually a composite feedforward-feedback system.

constraint. Let us suppose that a constraint is to be preferred if it requires no extra machinery beyond that needed to learn the unconstrained movement. This desideratum would rule out the minimum jerk principle proposed by Flash and Hogan (1985), because jerk is defined in Cartesian coordinates—a kinematic transformation of accelerations using the time derivative of the Jacobian matrix is required. It is much easier to implement a smoothness criterion on the Cartesian coordinates of the virtual trajectory; this simply requires a forward kinematic model, which is needed independently so that task constraints can be specified in Cartesian coordinates. In fact, this criterion also appears to give a better account of the data from arm movements, as has been argued by Flash (1987). The particular constraint suggested by Flash is a minimum jerk constraint; however, this constraint requires the acceleration of the virtual trajectory. Because this acceleration is not needed for learning the unconstrained movement, a constraint on a lower-order derivative of the virtual trajectory would be preferred. Tentative support for this argument is provided by data from monkey arm movements (see Fig. 2 in Hogan, 1984); the derivative of the inferred virtual trajectory is non-zero at the origin, which is inconsistent with a minimum jerk criterion.

Learning parameterized plans

Learned controllers must present a set of control parameters to higher-level processes so that the appropriate control surface can be retrieved in particular situations. These control parameters must constitute a reasonably small set, so that articulatory indeterminacies are not recapitulated on the input side of the controller. The parameters must also have a predictable relationship to the controller output, given that environmental situations are never exactly the same.

In the current approach, the plan vector is the interface between the controller and higher-level processes. The plan provides global information that designates the output to be generated; in dynamical terms, it instantiates a particular vector field. The relationship of “designation” is abstract—it is not necessary for the plan to encode the details of the controller output, because these details are already embedded in the weights of the controller subnetwork.

During learning, the error signals that are propagated through the controller subnetwork lead to internal adjustments that produce a more correct output in the context of the current state and plan vectors. Error signals also arrive at the plan units. These signals are the partial derivatives of the cost functional J with respect to the activations of the plan units; they therefore provide information about how the plan vector itself could be changed to minimize the cost functional. This information is potentially useful to encoding relationships between sequences, because sequences with similar subsequences tend to produce similar partial derivatives at the plan units. Thus, the possibility exists that appropriate constraints on plan representations are inherent in the transformed error signals.

To investigate this possibility a network with modifiable plan representations learned to control a six-degree-of-freedom planar manipulator with a single endpoint. To insure that the external input provided no information about how to form the plan representations,

the technique of using standard basis input vectors was borrowed from [14]. The network consisted of a set of four *local* plan units, one and only one of which was turned on for each of the four sequences being learned.⁵ These units were connected to a set of four *distributed* plan units, which were then connected to the remainder of the network. The weights between the local and distributed plan units were initially set to small random values, and were learned based on the error signals that were propagated back through the distributed plan units.

The results after 650 learning trials are shown in Figure 11. The figure shows the final performance of the network for each sequence and also displays the activations of the distributed plan units. These activations have reasonably clear interpretations, which are supported by a sensitivity analysis. The first distributed plan unit clearly encodes the vertical dimension of the workspace. This is useful given the translational articulatory degree of freedom in that direction. The second unit appears to take on three values that encode the average direction and amount of curvature of the arm. For the top-left and bottom-left sequences this curvature is large, in opposite directions, whereas for the rightmost pair of sequences, little curvature is required. For the third unit, investigation of the weights on the connections to the hidden units reveals that the unit serves to gate particular hidden unit clusters that vary in opposing directions in time. Thus, this unit encodes the direction of movement—the top-left and bottom-right sequences are counterclockwise, whereas the other two sequences are clockwise. Finally, the fourth distributed plan unit appears to redundantly encode the vertical dimension of the workspace. In general, these units developed representations that appear to be useful parameterizations of the sequences to be performed.

A final point that is demonstrated in the simulation is that control parameters do not need to encode the details of movement; rather, they need only make the relevant distinctions among the set of behaviors that the lower level network is capable of producing. The distinctions arise from errors present in the learning process that cannot be accounted for by contextual (state) variables. This systemic view of the notion of a control parameter is of course consistent with a venerable tradition in phonology—that of distinctive feature theory [17].

Conclusions

Among the numerous proposals for solving motor learning problems—direct inverse modeling [3, 24, 29, 35], reinforcement learning [4] and feedback-based inverse modeling [24]—the forward modeling approach is indubitably the most complex, requiring a separate learning process for the forward model as well as a differentiation procedure to make use of the forward model. Nonetheless, the approach has certain advantages (see [21] for a comparison of the various approaches). The current paper emphasizes one such advantage: The forward modeling approach allows intrinsic constraints on articulatory vectors and task vectors to be incorporated directly in the learning rule. By virtue of the “visible” character of artic-

⁵The local plan units should be thought of as placeholders for the output units of a higher level network.

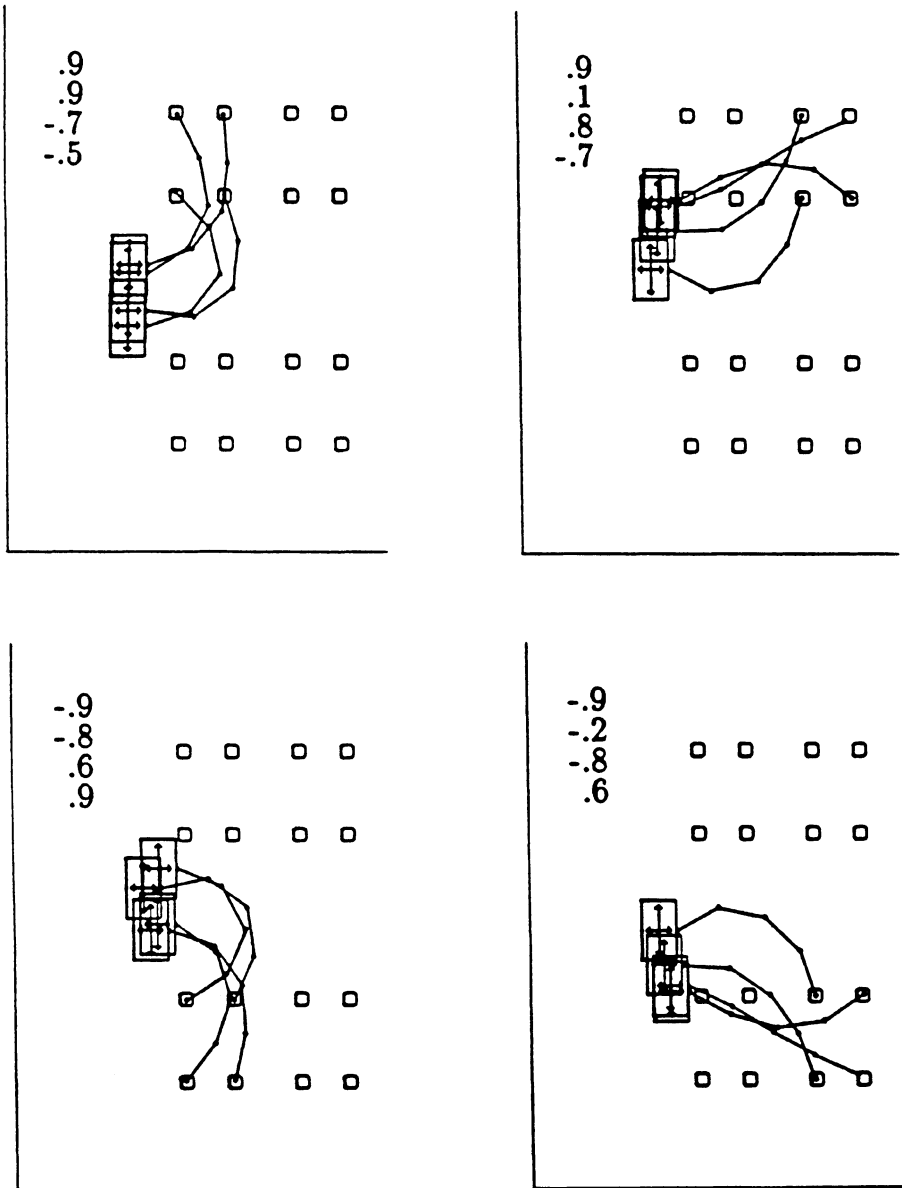


Figure 11: Activation patterns of the distributed plan units for each sequence and corresponding performance of the network.

ulatory units and task units, justification for the choice of particular intrinsic constraints can be sought in domain-specific terms.

References

- [1] Ackley, D. H., Hinton G. E., & Sejnowski T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9, 147-169.
- [2] Arbib, M. A. (1981). Perceptual structures and distributed motor control. In V. B. Brooks (Ed.), *Handbook of physiology – The nervous system II*. Bethesda, MD: American Physiological Society.
- [3] Atkeson, C. G. & Reinkensmeyer, D. (1988). Using associative content-addressable memories to control robots. *IEEE Conference on Decision and Control*. San Francisco, CA.
- [4] Barto, A. G. (1989). Connectionist learning for control. *Proceedings of the NSF Workshop on Application of Neural Networks to Robotics and Control*. Cambridge: MIT Press.
- [5] Berkenblit, M. B., Fel'dman, A. G. & Fucson, O. I. (1986). Adaptability of innate motor patterns and motor control. *Behavioral and Brain Sciences*, 9, 585-638.
- [6] Bizzi, E., Accornero, N., Chapple, W., & Hogan, N. (1984). Posture control and trajectory formation during arm movement. *Journal of Neuroscience*, 4, 2738-2745.
- [7] Eccles, J. C. (1977). *The Understanding of the Brain*. New York: McGraw-Hill.
- [8] Fel'dman, A. G. (1980). Superposition of motor programs. I. Rhythmic forearm movements in man. *Neuroscience*, 5, 544-548.
- [9] Fitts, P. M. (1964). Perceptual-motor skill learning. In A. W. Melton (Ed.), *Categories of Human Learning*. New York: Academic Press.
- [10] Flash, T. (1987). The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics*, 57, 257-274.
- [11] Flash, T. & Hogan, N. (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5, 1688-1703.
- [12] Greene, P. H. (1982). Why is it easy to control your arms? *Journal of Motor Behavior*, 9, 2-42.
- [13] Gurfinkel, V. S., & Levik, Y. S. (1979). Sensory complexes and sensorimotor integration. *Fiziologiya Cheloveka*, 5, 399-414.

- [14] Hinton, G. E. (1986). A network which learns distributed representations of concepts. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.
- [15] Hogan, N. (1984). An organising principle for a class of voluntary movements. *Journal of Neuroscience*, 4, 2745-2754.
- [16] Hogan, N. (1985). Impedance control: An approach to manipulation: Part I—Theory. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107, 1-24.
- [17] Jakobson, R., Fant, G., & Halle, M. (1951). *Preliminaries to speech analysis*. Cambridge, MA: MIT Press.
- [18] Jordan, M. I. (1988). *Supervised learning and systems with excess degrees of freedom*. (COINS Tech. Rep. 88-27). Amherst, MA: University of Massachusetts, Computer and Information Sciences.
- [19] Jordan, M. I. (1990). Motor learning and the degrees of freedom problem. In M. Jeannerod, ed. *Attention and Performance, XIII*. Hillsdale, NJ: Erlbaum.
- [20] Jordan, M. I. (in press). Serial order: A parallel, distributed processing approach. In J. L. Elman and D. E. Rumelhart, (Eds). *Advances in Connectionist Theory: Speech*. Hillsdale, NJ: Erlbaum.
- [21] Jordan, M. I. & Rosenbaum, D. A. (1990). Action. In M. I. Posner (Ed.), *Foundations of Cognitive Science*. Cambridge, MA: MIT Press.
- [22] Jordan, M. I. & Rumelhart, D. E. (1990). Supervised learning with a distal teacher. Submitted to: *Cognitive Science*.
- [23] Kawato, M. (1990). Computational schemes and neural network models for formation and control of multijoint arm trajectory. *Proceedings of the NSF Workshop on Application of Neural Networks to Robotics and Control*. Cambridge: MIT Press.
- [24] Kawato, M., Furukawa, K., & Suzuki, R. (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57, 169-185.
- [25] Kelso, J. A. S. & Holt, K. G. (1980). Exploring a vibratory systems analysis of human movement production. *Journal of Neurophysiology*, 43, 1183-1196.
- [26] Kent, R. D. & Minifie, F. D. (1977). Coarticulation in recent speech production models. *Journal of Phonetics*, 5, 115-117.
- [27] Kiparsky, P. (1975). Comments on the role of phonology in language. In J. F. Kavanagh & J. E. Cutting (Eds.), *The Role of Speech in Language*. Cambridge: MIT Press.
- [28] Koditschek, D. E. (1984). Natural control of robot arms. *IEEE Proceedings of the 23rd Conference on Decision and Control*. pp. 733-735.

- [29] Kuperstein, M. (1987). Adaptive visual-motor coordination in multijoint robots using parallel architecture. *IEEE Conference on Robotics and Automation*, pp. 1595-1602.
- [30] LeCun, Y. 1985. A learning scheme for asymmetric threshold networks. *Proceedings of Cognitiva 85*. Paris, France.
- [31] Lieberman, P. (1975). *On the origins of language: An introduction to the evolution of speech*. New York: Macmillan.
- [32] Lindblom, B. (1983). Economy of speech gestures. In P. F. MacNeilage (Ed.), *The Production of Speech*. New York: Springer-Verlag.
- [33] Lindblom, B., Lubker, J., & Gay, T. (1979). Formant frequencies of some fixed-mandible vowels and a model of speech motor programming by predictive simulation. *Journal of Phonetics*, 7, 147-161.
- [34] Martinet, A. (1968). Phonetics and linguistic evolution. In B. Malmberg, (Ed.), *Manual of Phonetics*. Amsterdam: North-Holland.
- [35] Miller, W. T. (1987). Sensor-based control of robotic manipulators using a general learning algorithm. *IEEE Journal of Robotics and Automation*, 3: 157-165.
- [36] Mussa-Ivaldi, F. A., McIntyre, J., & Bizzi, E. (1988). Theoretical and experimental perspectives on arm trajectory formation: A distributed model for motor redundancy. In E. Clementi & S. Chin (Eds.) *Biological and Artificial Intelligence Systems*. Escom.
- [37] Nguyen, D., & Widrow, B. (1988). Personal communication.
- [38] Öhman, S. E. G. (1966). Coarticulation in VCV utterances: spectrographic measurements. *Journal of the Acoustical Society of America*, 39, 151-168.
- [39] Parker, D. 1985. *Learning-logic*. (Tech. Rep. TR-47). Cambridge, MA: MIT Sloan School of Management.
- [40] Pineda, F. J. (1987). Generalization of backpropagation to recurrent and higher order neural networks. *Proceedings of the IEEE Conference of Neural Information Processing Systems*, Palo Alto: Morgan Kaufmann.
- [41] Polit, A. & Bizzi, E. (1978). Processes controlling arm movements in monkeys. *Science*, 201, 1235-1237.
- [42] Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Volume 1*. Cambridge, MA: MIT Press.
- [43] Salisbury, J. K. (1980). Active stiffness control of a manipulator in Cartesian coordinates. *Proceedings of the 19th IEEE Conference on Decision and Control*, 95-100.

- [44] Saltzman, E. L. & Kelso, J. A. S. (1987). Skilled actions: A task-dynamic approach. *Psychological Review*, 94, 84-106.
- [45] Saltzman, E. L. & Munhall, K. G. (1988). A dynamical approach to gestural patterning in speech production. Submitted to *Ecological Psychology*.
- [46] Uno, Y., Kawato, M., & Suzuki, R. (1987). *Formation of optimum trajectory in control of arm movement—minimum torque-change model*. (Japan IEICE Tech. Rep. MBE 86-79). Osaka, Japan.
- [47] Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Unpublished doctoral dissertation, Harvard University.
- [48] Werbos, P. (1987). Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research. *IEEE Transactions on Systems, Man, and Cybernetics*, 17, 7-20.
- [49] Widrow, B. & Stearns, S. D. (1985). *Adaptive signal processing*. Englewood Cliffs, NJ: Prentice-Hall.
- [50] Williams, R. J. & Zipser, D. (1988). *A learning algorithm for continually running fully recurrent neural networks*. (Tech. Rep. ICS 8805). University of California, San Diego, Institute for Cognitive Science.

Proposal for a Pattern Matching Task Controller for Sensor-Based Coordination of Robot Motions

Martin Brooks

Center for Industrial Research, Box 124 Blindern, 0314 Oslo 3, Norway

Abstract. This paper addresses the problem of endowing a robot manipulator with the ability to skillfully manipulate objects in its environment. We propose applying pattern matching techniques to sensory data as a mechanism for starting and stopping simple compliant motions. We describe a pattern matching task controller's role in a control hierarchy having a human supervisor at the topmost level. Control of the manipulator is shared between the supervisor and the task controller, allowing the task controller to operate the manipulator autonomously for periods of a few seconds during execution of previously rehearsed tasks. This autonomous control consists of execution of a sequence of predetermined motion states, i.e. simple compliant movements and/or applications of force, where the transitions between motion states are triggered by the pattern matcher in response to highly processed sensor data. The contents of the motion states and the discriminants used by the pattern matcher are learned during an earlier training phase.

1. How Can We Develop Skilled Manipulators?

This paper addresses the problem of endowing a robot manipulator with the ability to skillfully manipulate objects in its environment. The Random House Dictionary defines skill as "*the ability, coming from one's knowledge, practice, aptitude, etc., to do something well*". Skill is a task-related attribute. The concept "skilled manipulator" is only meaningful if we say what the manipulator is skilled at. The question then is: What mechanisms must a manipulator have in order to be able to become skilled at something, and what procedures must be followed to obtain such skill?

Today's robot task descriptions are usually either a robot program listing a sequence of motions, or playback of a continuous path recorded under the control of a human operator. Compliance, force commands, and hybrid position-force control have been added to robot programs [9] [10], and path recordings have been generalized to arbitrary sensors [11], but none of these by itself has significantly improved the manipulative capabilities of robots. We believe that robot programs need an inherently richer language for utilizing sensory information. We would like to replace statements like `IF force > 13 THEN ...` with statements like `IF match(data, pattern) THEN ...`, where `data` is the current multidimensional sensory state and `pattern` is learned during trial executions.

Teleoperated manipulators are operated by a person using a master arm, joystick, or similar device. Forces acting on the manipulator may be reflected back to the operator's hand [30] or sensor data may be presented on a screen [28] [29] [30]. Cameras may provide views of the manipulator's end-effector. The manipulator's skill level is directly dependent on the operator's motions. Teleoperation requires a high degree of operator concentration. We

believe that the manipulator skill level can be increased, and the load on the operator decreased, by sharing control of the manipulator between the operator and a computer which responds to sensor data. This is an example of *supervisory control* [12]. We propose a particular architecture where short motion sequences may be automatically carried out by the manipulator, using sensor data to coordinate the individual motions.

The task controller proposed in this paper could be used either at the task execution level in a control hierarchy for an autonomous robot [23], [26], or as an extension of teleoperation to supervisory control [12]. The two formulations are made equivalent by considering the supervisor as a replacement for the higher levels in the hierarchy.

One of our primary goals is to identify principles for skilled manipulation that do not require intelligence on the part of the task controller. We expect that robotic manipulative skill will be easier to develop than robotic intelligence. We would like to implement skill mechanisms and gain experience with them, so that they can become a solid stepping stone in the path towards intelligent systems.

Some degree of intelligence is generally considered a prerequisite for autonomous robot operation. If this is the case, then a skilled but not intelligent robot cannot be fully autonomous. For this reason we prefer to see our task controller in the supervisory control context. Supervisory control avoids the need for artificial intelligence by converting machine intelligence problems to man-machine interface problems.

Another primary goal is to construct sensor-based control mechanisms that are unconditionally stable despite the use of *non-located* sensors. A *located* sensor directly measures a controlled state at the site of actuation, e.g. actuator position, velocity, or torque. A *non-located* sensor measures a controlled state distal to the site of actuation, e.g. a wrist force/torque sensor. Non-located sensors contribute to instability because the manipulator's physical dynamics intercede between the actuator and sensor [8]. The scheme presented in this paper makes use of non-located sensors without incurring stability problems because the sensor data is not used as feedback at the servo-control level, but instead is used to trigger state transitions in a finite state machine at the rate of at most one or two transitions per second.

In this paper we propose use of pattern matching techniques applied to sensor states as a mechanism for starting and stopping simple compliant motions and applications of force. A *task* is executed as a sequence of such motions. An example of a task is "prying open a door with a lever". The end of the lever must be inserted into the crack; force is applied until the door begins to open or until the tool begins to bend or slip. Another example is "giving a tool to a person". A robot is holding a hammer in its gripper. The person grasps the hammer and begins to draw it away. The robot responds by releasing its grip. Other classes of tasks include grasping and regrasping, rubbing, scraping, insertion, screwing, striking, pinching, turning a dial, throwing a switch, bending, folding, winding, wringing, and so on.

Our formulation of "task" is quite low-level and procedural. It corresponds to task execution at level 4 in the NASREM architecture [23]. We visualize tasks as finite state machines, where the states are *motion states*, i.e. simple compliant motions or applications of force, and the transitions between states are guarded by *sensory predicates* of the form `match(data, pattern)`. See figure 1. A task execution is a path through this graph. A

typical task execution will consist of ten or so motion states, with state transitions no more often than one or two times a second; i.e. the total duration will be ten or twenty seconds.

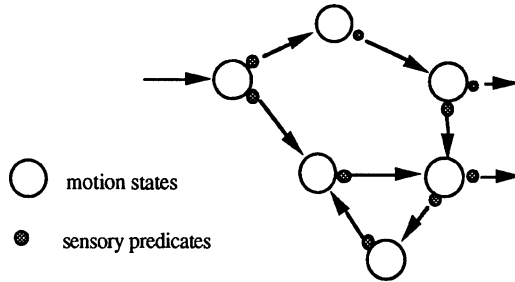


Figure 1: A task as a finite state machine. During task execution the path through the graph and the timing of the transitions is chosen according to the sensory states.

In fact this formulation is an oversimplification of manipulative skill in two respects: (1) A manipulator with many degrees of freedom, e.g. an arm with an articulated hand, may carry out several such motion sequences concurrently, and (2) not all the motions making up skilled manipulation are the result of this type of representation; in addition some may be reflex motions and others may be under direct control of intelligent problem solving.

The task controller described in this paper is essentially the same as Albus' proposal in [21, chap. 9], except that we are not concerned with the details of one specific pattern matching method (i.e. CMAC) or with the creation of extended control hierarchies [23]. Asada and Yang [13] have implemented a similar system for robotic deburring, where sensor data is used to control transitions between states defining several levels of grinding force and repeated passes along segments with large burrs. Hannaford and Lee [15] describe a related idea, where a hidden Markov model defines the states and controls the transitions between them.

2. What is Manipulative Skill?

Figure 2 shows manipulative skill in relation to a hierarchy of sensorimotor control abilities. Skilled manipulation overlaps three sensorimotor abilities: reflex control [14] [33] [27], learned motion sequences [13] [15] [21] [25], and intelligent movement [21] [31] [26]. The scheme described in this paper concerns learned motion sequences. As shown in figure 1, a motion sequence is a path through a graph. What is learned is the structure and contents of the graph, including the sensory predicates. During task execution the choice and timing of the path through the graph is determined according to the sensory predicates.

A task is carried out as a sequence of interactions between the manipulator and its environment, or more generally, several *concurrent* sequences of interactions. Some of these interactions may be reflexive. Some may be learned motion sequences. Other interactions may be the result of intelligent on-the-fly planning. Skilled manipulation uses all

three modes. The need for intelligent motion depends on the particular task and the degree to which task execution proceeds similarly to previous executions of the same task. We assume that one wants to use intelligence as little as possible, and would rather execute the task using a learned motion sequence. This frees computational resources for applying intelligence to other problems while the task is being executed.

Manipulative Skill	Sensorimotor Ability	Methods	Dominant Issues	Status
	actuator control	servo & model-based control, collocated sensors for controlled actuator states e.g. position, velocity, torque	accuracy tracking a time-varying command	works excellently
	endpoint position, force, and compliance control	kinematic & dynamic models, redundancy resolution, trajectory generation, non-collocated sensors, e.g. wrist force/torque sensing	dynamic models, use of redundancy, stability in contact with unmodelled environment	progress being made, e.g. [1] [2] [3] [4] [5]
	reflex control	direct control loops around sensors and actuators	identifying appropriate applications	walking [27] [14]; gripping [33]
	learned motion sequences	finite state models, pattern matching, supervisory control	range of applicability, pattern matching techniques, definition of features for pattern matching, operator interfaces	some experimental implementations [15] [25]
	intelligent movement	multilevel hierarchical control	perception, planning	basic research [21] [26] [31]

Figure 2: Manipulative skill and robotic sensorimotor abilities.

Interaction with the environment is comprised of combinations of motion and application of force. The problems are *where*, *how much*, and *when*. With respect to learned motion sequences, we believe that *when* is the problem to concentrate on. *When* includes the onset, duration, and termination of each interaction. *When* coordinates each interaction with the state of the environment and with other concurrent manipulations. In the graph describing a task (figure 1) the *where*'s and *how much*'es are contained in the motion states; *when* is determined by the sensory predicates. The motion states are predetermined by an intelligent planning process, and are not changed without further application of such intelligence. During task execution a motion sequence results from a conditional choice of transitions leaving the current state. These choices are fixed; it is the timing of the transition that is variable. One could separate these issues by considering only *linear task graphs*, as in figure 3. In linear tasks graphs *when* is the only issue, since there is no choice of *what*

or *how much*. However, more general graphs appear to be appropriate; they include possibilities for exception handling and conditional outcomes of motions and force applications.



Figure 3: A linear task graph.

3. Compliance

Contact with the environment is a fundamental problem for robotics. A key concept is manipulator *compliance*, i.e. the ability to move in response to externally applied forces. Manipulator compliance plays a central role in our formulation of manipulative skill. Compliance can be equated with safety because it reduces contact forces between the manipulator and the environment. Compliance is not a part of manipulative skill, but it is a prerequisite for interaction with partially known environments, and therefore a prerequisite for manipulative skill.

Development of manipulator and controller designs providing stable endpoint compliance during contact with unmodelled environments is a central area of robotics research [1] [2] [3] [4] [5] [6] [7] [8]. Developments related to Hogan's impedance control concept [1] [2] [3] [4] seem promising, especially in conjunction with manipulators designed expressly for impedance control [2].

An important aspect of controlled compliance is the passive compliance of the manipulator, i.e. the ability to move the end-effector by applying forces, backdriving the actuators. Lawrence [6] shows that for typical industrial manipulators using a non-collocated force sensor (e.g. wrist force-torque sensor), only a limited range of compliances can be stably obtained. A non-collocatedly controlled, passively stiff manipulator in contact with a stiff environment becomes unstable unless it is heavily damped. Conversely, a non-collocatedly controlled, passively soft manipulator in contact with a soft environment becomes unstable unless it is heavily damped.

Stability limitations can be avoided by going outside the class of industrial manipulators, adding extra actuators where necessary in order to avoid control via non-collocated sensors. Sharon, Hogan, and Hardt [2] show that a macro/micro manipulator configuration (i.e. a small robot carried by a larger one) can stably perform high bandwidth position and force control. Colgate and Hogan [4] show that a controlled system will be stable in contact with any passive environment exactly when the controlled system appears to be passive as seen by the environment.

These results suggest that a robotic manipulator intended for skilled manipulation should have a macro/micro structure, i.e. arms with fingers, and that each sub-manipulator should be passively compliant. The macro/micro structure provides stable high bandwidth control over a wide range of impedances; the passive compliance gives us fail-safe operation in circumstances where the sub-manipulators cannot function together.

4. The Pattern Matching Task Controller in a Control Hierarchy

The pattern matching task controller is presented in the context of *supervisory control* [12]. This means that control of the manipulator is shared between two entities: a *supervisor* and the controller itself. We take the supervisor to be human, but in principle it could be an artificial intelligence. In addition we introduce a *sensory processor*. These fit into a three-level control hierarchy as shown in figure 4.

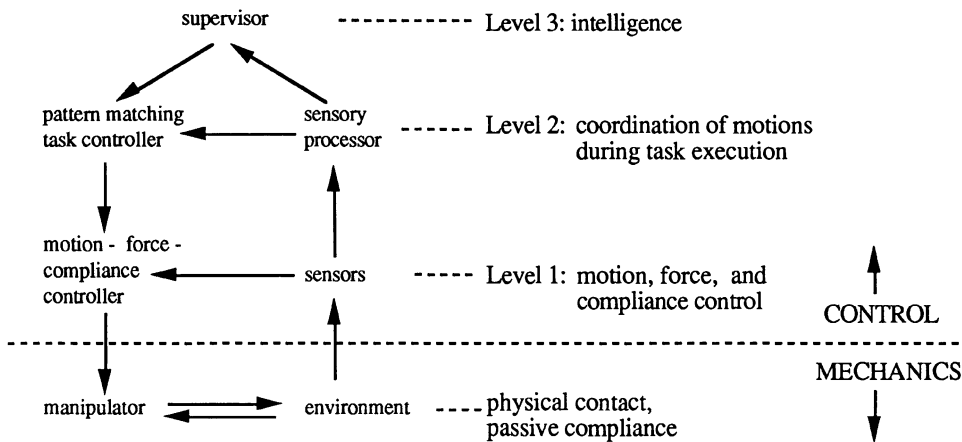


Figure 4. Three level hierarchical control.

The supervisor, level 3, might be interfaced to the system through a combination of joysticks, a master arm, video and computer screens, etc. The supervisor shares control of the manipulator with levels 1 and 2 to various degrees.

Level 2 provides task control. Level 2 corresponds to NASREM's level 4 [23]. The sensory processor takes in low-level sensor data and builds various representations based on processes of selection, filtering, estimation, and fusion. Advanced sensory interpretation such as speech, written language, and higher vision are not handled in the sensory processor. The intention is that the sensory processor should include only processing that can clearly be computed with algorithms of signal processing, control, and estimation theory, and should not include processing requiring cognitive evaluation and interpretation.

Level 1 executes compliant motions and applications of force. Level 1 includes NASREM's first two levels, servo and primitive. NASREM's level 3, e-move, which does obstacle avoidance, is pushed up into the supervisor's domain since it appears to require intelligence.

For simplicity we have not explicitly represented reflex control in the hierarchy of figure 4. It would have appeared between levels 1 and 2. Reflex control has been used for robotic gripping [33] and legged locomotion [14] [27]. It is certainly important for manipulation, but we have omitted it to focus on level 2.

5. Operational Modes

The hierarchical supervisory control system shown in figure 4 can be operated in three modes: direct operator control, automatic task execution, and task learning mode.

In direct operator control mode the supervisor uses a joystick, a master arm, or a similar interface device to command the level 1 motion - force - compliance controller. The task controller (level 2) is bypassed. This mode is basically like standard teleoperation. The joystick or master arm may be force-reflecting [30], or forces and other sensor data may be presented graphically [28] [29] [30]. Compliance control adds complexity not found in today's teleoperators; task-related choice of manipulator impedance is being investigated [32].

During automatic task execution the supervisor is inactive and the task controller (level 2) runs the manipulator. The tasks we have in mind are short (10 - 20 seconds), consisting of a sequence of ten or so motion states. Examples are gripping, prying, twisting, etc. Each motion state is a command down to the level 1 motion - force - compliance controller. The task controller makes transitions between motion states when it recognizes previously learned patterns in the sensory processor's output. Automatic task execution is initiated by the supervisor and can be aborted at any time simply by taking over control of the manipulator. During automatic task execution the supervisor's controls (i.e. joystick, master arm, etc.) move as though he had been commanding the movement. He can keep his hands lightly on the controls as execution proceeds.

In task learning mode the supervisor supplies the structure of the task description to the task controller and then repeatedly executes the task as in direct control mode. The task description might be a graph as in figure 1. During the directly controlled execution the supervisor continually indicates which motion state obtains. The task controller builds a motion - force - compliance command for each motion state based on the supervisor's observed behavior. The sensory data at the times the supervisor makes transitions between motion states is used as input to pattern learning routines which build discriminants for each transition. When sufficient information has been collected the task controller indicates that it is prepared to execute the task automatically. The supervisor keeps his hands on the controls during these attempted automatic executions, adjusting their motion states and the timing of their transitions until performance is satisfactory.

6. Controller Design

The algorithmic design of the pattern matching task controller is essentially an event-driven loop that gets traversed once for every transition between motion states. A pattern matcher continually monitors the output of the sensory processor, generating a state transition event whenever such a pattern in the sensor data is recognized. This design is essentially the same as that of Asada and Yang [13].

Two key design issues are the internals of the sensory processor and the pattern matcher. The sensory processor provides the feature data used by the pattern matcher based on processes of selection, filtering, estimation, and fusion. The sensors used to obtain the raw

data and the subsequent processing of this data will determine which types of patterns can be successfully learned and recognized by each particular type of pattern matcher. At this point we do not want to commit to any particular sensory representation or pattern matching style. Determination of the most appropriate solutions is an experimental matter and will be the focus of future research.

7. Issues for Pattern Matching Control

In this section we mention four key issues for pattern matching control: feature selection, pattern matching techniques, time-advanced recognition, and hidden Markov models.

Feature selection: The sensory processor supplies processed sensor data, i.e. features, to the pattern matcher as shown in figure 4. The issues are: Which features are appropriate for a specific task? Is there a "universal" set of features?

Pattern matching techniques: The pattern matcher must build discriminants from training data and then apply these discriminants in real time during task execution. Available techniques include statistical methods, neural nets, and rule-based systems (a review and comparison is found in Pao [16]) as well as hashing-based techniques like Albus' CMAC [21].

Time-advanced recognition: During task learning the supervisor directly controls the manipulator while telling the pattern matching controller which motion state currently obtains and when the transitions between motion states have occurred. The pattern matcher builds discriminants for the sensor patterns indicating transitions based on the sensory state at the transition times indicated by the supervisor. Due to information processing and physical delays it may be necessary to associate each transition with the sensory state obtaining a short time *before* the supervisor's indicated times.

Hidden Markov models: Our proposed scheme focuses on the sensor patterns indicating transition between motion states. Hannaford and Lee [15] use hidden Markov models (HMM) based on the sensory states that obtain *during* each motion state together with independent probabilities of transitions between states. The HMM allows them to parse the sensor data stream into a compact state-based transcription of the operator's actions. It seems that HMM's might also be useful for our proposed pattern matching controller.

8. Biological Analogy

The pattern matching task controller resembles a speculative interpretation of the function of the lateral cerebellum in humans and other vertebrates. We have based this interpretation on reviews and summaries of cerebellar structure and function in [18] [19] [20] [22] [24].

The Albus/Marr theory of cerebellar function [22] [24] views the cerebellar cortex as an adaptive pattern recognizer. The cerebellar cortex is divided into many microzones [19, Chap 15], each of whose output presumably corresponds to an elemental movement. Patterns consist of a combination of sensory and limbic (i.e. internal, goal-related) data [20, Chap. 13.3]. The sensory data arriving at the lateral cerebellum comes from the cerebral

cortex, i.e. it is already highly processed. In planar arm-motion experiments with monkeys [20, Chap. 13.3], the output of the lateral cerebellum (via the dentate nucleus) gives a strong response just prior to starting and stopping of task-related movements. These responses are forwarded to the motor and premotor cortex, where they presumably initiate what eventually becomes the muscle contractions that start and stop the movements.

Our interpretation of these results assigns the lateral cerebellum's function to task control in a manner similar to the proposed pattern matching controller. A task's description is downloaded into the lateral cerebellum just prior to execution. (For voluntary motions there is a period of about .8 seconds between the onset of brain activity and muscle activity during which such a downloading could take place [20, Chap 6.1].) The task representation is a finite-state graph, however each motion state is spatially distributed over the microzones whose elemental movements collectively effect the motion state. Each microzone acts independently as a recognizer for patterns that should trigger its elemental movement during task execution. This recognition is slightly time-advanced in relation to the resulting movement.

9. Summary

In this paper we have outlined a proposal for a pattern matching task controller. We have described the task controller's role in a hierarchy having a human supervisor at the topmost level. Control of the manipulator is shared between the supervisor and the task controller, allowing the task controller to operate the manipulator autonomously for periods of a few seconds during execution of previously rehearsed tasks. This autonomous control consists of execution of a sequence of predetermined motion states, i.e. simple compliant movements and/or applications of force, where the transitions between motion states are triggered by a pattern matcher applied to highly processed sensor data. The contents of the motion states and the discriminants used by the pattern matcher are learned during an earlier training phase.

The purpose of the pattern matching task controller is to enable a robot manipulator to carry out small short manipulative tasks whose successful execution is dependent on coordination of the robot's motions with the state of the environment. We equate this ability with "manipulative skill", and postulate that the combination of these learned motion sequences together with intelligent motion and reflex control are sufficient to endow a manipulator with such skill. We have discussed two design goals for the pattern matching task controller: (1) To separate "skill" from "intelligence", and (2) unconditional stability despite the use of non-collocated sensors.

This paper has only briefly mentioned the technical issues stemming from the proposed controller. These are the subject of ongoing and future research.

Bibliography

1. Neville Hogan
Impedance Control: An Approach to Manipulation: Part I - Theory
J. Dynamic Systems, Measurement, and Control, 107:1-7, March 1985
2. Andre Sharon, Neville Hogan, and David E. Hardt
Controller Design in the Physical Domain (Application to Robot Impedance Control)
IEEE Conf. Robotics & Automation, Scottsdale, Arizona, USA, May 1989
3. Neville Hogan
Controlling Impedance at the Man/Machine Interface
IEEE Conf. Robotics & Automation, Scottsdale, Arizona, USA, May 1989
4. Ed Colgate and Neville Hogan
An Analysis of Contact Instability in Terms of Passive Physical Equivalents
IEEE Conf. Robotics & Automation, Scottsdale, Arizona, USA, May 1989
5. Chae H. An, Christopher G. Atkeson, John M. Hollerbach
Model-Based Control of a Robot Manipulator, MIT Press, 1988
6. Dale A. Lawrence
Impedance Control Stability Properties in Common Implementations
IEEE Int. Conf. Robotics & Automation, Philadelphia PA, USA, 1988
7. Steven D. Eppinger and Warren P. Seering
On Dynamic Models of Robot Force Control
IEEE Int. Conf. Robotics and Automation, San Francisco, CA , USA, 1986
8. Steven D. Eppinger and Warren P. Seering
Understanding Bandwidth Limitations in Robot Force Control
IEEE Int. Conf. Robotics and Automation, Raleigh, NC , USA, 1987
9. M. H. Raibert and J. J. Craig
Hybrid Position/Force Control of Manipulators
J. Dynamic Systems, Measurement, and Control, vol. 102, June 1981
10. M. T. Mason
Compliance and Force Control for Computer Controlled Manipulators
IEEE Trans. Systems, Man, and Cybernetics, Vol SMC-11, no. 6, June 1981
11. Gerhard Hirzinger
Robot Systems Completely Based on Sensory Feedback
IEEE Trans. Industrial Electronics, Vol IE-33, no. 2, May 1986
12. T. B. Sheridan
Teleoperation Scorecard: Déjà vu and Really New
Proc. Int. Symp. Teleoperation and Control, 3 - 9, July 1988
13. Haruhiko Asada and Boo-Ho Yang
Skill Acquisition from Human Experts Through Pattern Processing of Teaching Data
IEEE Conf. Robotics & Automation, Scottsdale, Arizona, USA, May 1989
14. R. Tomocic and G. A. Bekey
Robot Control by Reflex Actions
IEEE Int. Conf. Robotics and Automation, San Francisco, CA , USA, 1986
15. Blake Hannaford and Paul Lee
Hidden Markov Model Analysis of Force/Torque Information in Teleoperation
Proc. 1st Int. Symp. Exp. Robotics, V. Hayward and O. Khatib, eds.,
Springer Verlag, Montreal, June 1989
16. Yoh-Han Pao
Adaptive Pattern Recognition and Neural Networks, Addison-Wesley, 1989
17. Martin Brooks
The DataGlove as a Man-Machine Interface for Robotics
The Second IARP Workshop on Medical and Healthcare Robotics, Newcastle upon Tyne, UK
5-7 September 1989
18. Eric R. Kandel and James H. Schwartz
Principles of Neural Science
Second Edition, Elsevier Science Publishing Co., New York, USA, 1985
19. Masao Ito
The Cerebellum and Neural Control
Raven Press, New York, USA, 1984

20. Vernon B. Brooks
The Neural Basis of Motor Control
Oxford University Press, New York, USA, 1986
21. James S. Albus
Brains, Behavior, and Robotics
Byte Books, Subsidiary of McGraw-Hill, H.H., USA, 1981
22. James S. Albus
A Theory of Cerebellar Function
Mathematical Biosciences 10, 25-61, 1971
23. James S. Albus, Harry G. McCain, and Ronald Lumia
NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)
NIST Technical Note 1235, National Institute of Standards and Technology,
Gaithersburg, MD 20889, USA, July 1987
24. David Marr
A Theory of Cerebellar Cortex
J. Physiol. 202, 437-470, 1969
25. Bruno Dufay and Jean-Claude Latombe
An Approach to Automatic Robot Programming Based on Inductive Learning
LIFIA, BP 68, 38402 Saint-Martin-d'Heres Cédex, France, RR no. 433, February 1984
26. A. Meystel
Nested Hierarchical Intelligent Module for Automatic Generation of Control Strategies
in: Languages for Sensor-Based Control in Robotics, NATO ASI Series, Vol. F29
U. Rembold and K. Hormann, eds., Springer-Verlag 1987
27. Rodney A. Brooks
A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network
IEEE Conf. Robotics & Automation, Scottsdale, Arizona, USA, May 1989
28. Ming Ouh-young, David V. Beard, and Frederick P. Brooks, Jr.
Force Display Performs Better than Visual Display in a Simple 6-D Docking Task
IEEE Conf. Robotics & Automation, Scottsdale, Arizona, USA, May 1989
29. Won S. Kim and Lawrence W. Stark
Cooperative Control of Visual Displays for Telemanipulation
IEEE Conf. Robotics & Automation, Scottsdale, Arizona, USA, May 1989
30. A.K. Bejczy and B. Hannaford
Man-Machine Interaction in Space Telerobotics
Proc. Int. Symp. Teleoperation and Control, July 1988
31. T. Lozano-Pérez, M.T. Mason, and R.H. Taylor
Automatic Synthesis of Fine-Motion Strategies for Robots
Proc. Int. Symp. Robotics Research, Bretton Woods, NH, USA, Sept. 1984
32. G. J. Raju
Adjustable Impedance Control for Master-Slave Teleoperators
MIT Man-Machine Systems Lab Memo, 1987
33. R. Tomovik, G. Bekey, and W. Karplus
A Strategy for Grasp Synthesis with Multifingered Robot Hands
Proc. IEEE Int. Conf. Robotics and Automation, Raleigh, NC 1987

SENSORY-MOTOR MAPPING WITH A SEQUENTIAL NETWORK

Lina Massone and Emilio Bizzi

Dept. of Brain and Cognitive Sciences
Massachusetts Institute of Technology
77 Massachusetts Avenue - Cambridge Ma 02139

ABSTRACT

This paper deals with the problem of representing and generating unconstrained aiming movements of a limb by means of a sequential neural network. Targets are specified to the network as sensory stimuli; for each stimulus the network produces a time trajectory of a redundant limb from a starting posture towards the corresponding target. The network was trained using a bell-shaped velocity profile on the trajectories, which is a recurring feature of movements performed by biological systems. We performed a number of experiments, both during and after learning. Our results show that: (i) the task can be learned by a three-layer sequential network; (ii) the network successfully generalizes in trajectory space and adjusts the velocity profiles properly; (iii) the same task cannot be learned by a linear network; (iv) the model is robust to noise on the input signals; (v) the network exhibits attractor-dynamics properties; (vi) the network is able to solve the motor-equivalence problem. A key feature of this work is the fact that the neural network was coupled to a mechanical model of a limb in which muscles are represented as springs. This representation made it possible to deal with the redundancy of the motor apparatus.

1 Introduction

This paper deals with the problem of representing and generating unconstrained aiming movements of a limb by means of a neural network architecture [29, 30].

Aiming movements are present in biological systems at different levels of complexity, from accurately planned movements to reflexes [13]. The present work focuses on unconstrained limb movements elicited by sensory stimulation. They are meant to mimic the wiping movements made by the leg of spinal frogs when the frog's skin is stimulated by an irritant [3, 14]. Scratch reflexes of spinal cats [39] represent another example of this class of movements. Opto-electrical recordings of frogs' wiping movements [14] show that the motor strategy remains basically the same in both intact and spinal animals. This result suggests that the basic motor programs for this particular task are generated at the spinal cord level and not explicitly planned by higher brain structures. Starting from this observation, we adopted a non-hierarchical neural network to represent such movements.

The neural network's task in this work involved generating a trajectory of a limb from a starting posture toward a target specified in terms of a sensory stimulus. Hence, the network performed a sensory-motor transformation. Aiming movements were assumed to be planar (as

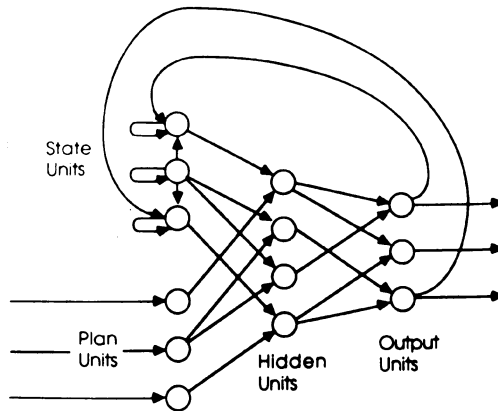


Figure 1: The basic architecture of Jordan's sequential networks.

in the aiming phase of the movement made by the frog when it wipes its back), but there is no theoretical limitation to the dimensionality the network could deal with.

In the present work, the experimenters used a bell-shaped velocity profile for training trajectories. This profile is a commonly recurring feature of movements performed by biological systems [32, 1, 2, 19, 11]. The duration of movements was assumed to be constant. As a consequence, when the neural network was asked to generalize, it was required not only to generate the correct path of the limb towards the target but also to adjust the velocity profile accordingly.

The paper investigates some features of the learning task, the generalization capabilities of the sequential network and the organization of the resulting connectivity pattern. Then we present some experiments performed with the trained network that demonstrate the robustness of the model, its dynamic properties and its implicit capability of solving the motor-equivalence problem.

2 Sequential Networks

Neural networks have been frequently applied to the robotic and motor control fields [24, 25, 21, 31, 8, 38, 26, 6, 27]. The architecture that we chose to produce time sequences is a sequential network of the type proposed by Jordan [20]. Figure 1 shows the basic structure of such network; it is composed of two arrays of input units (plan units and state units), one array of hidden units and one array of output units. Due to the recurrent connections from output units to state units, this network is able to produce sequences of output signals. The output units convey signals to the state units thereby changing the latter's level of activation. In turn, the state units provide a time-varying input to the layered network which learns the sequences. The self-connections to the state units make the next state of the network a function of the whole past history. The other input to the layered network derives from the plan units, which are activated by the external stimuli. The activation of the plan units remains constant within a given sequence but varies

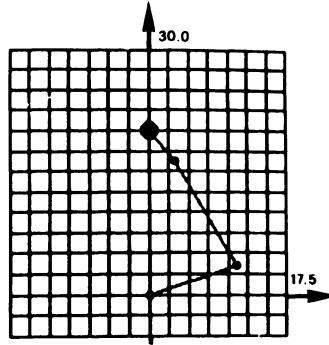


Figure 2: A portion of the limb workspace discretized with a 15x15 grid. The grid step is 2.5. The limb is in the starting posture and the origin of the coordinates is in the shoulder. The lengths and proportions of the three joints are those of a monkey arm.

between sequences to allow different sequences to be learned by the same network. The network is a dynamic system in which both output functions and state functions change as the weights in the forward path of the network are modified through learning. The remainder of this section shows how the aiming task was formulated in terms of sequential networks.

The output units drove a redundant three-joint limb which moved from the initial posture to the target. The limb was schematically modeled with four pairs of antagonist muscles: the shoulder flexor and extensor, the double joint flexor and extensor, the elbow flexor and extensor, the wrist flexor and extensor. Muscles were represented as springs according to a model that is described in Section 3. Each output unit activated a muscle; output units were considered as motoneurons. Hence, the time sequence generated by the network was encoded in muscle space (not in joint or end-point space). It is worth noting that generating a sequence of postures in muscle space (eight degrees of freedom) which implements a planar trajectory with two degrees of freedom is an ill-posed problem with an infinity of solutions.

Plan units contained a representation of the sensory stimulus. It was assumed that the limb workspace and the limb itself could be measured in the same body-centered reference frame (as in the aiming phase of the frog's wiping movement to the back). Consequently, there was no coordinate-transformation problem ¹. A portion of the limb workspace was discretized with a 15x15 pixel grid as shown in Figure 2; Figure 2 also shows the initial posture of the limb for all aiming movements. The choice of the grid step (i.e. the resolution over the input space) was arbitrary, but we repeated the same learning and generalization procedures for a different grid step with the purpose of understanding how a change in resolution might affect the performance of the network (see Section 3). The stimulus was coded as a narrow gaussian distribution centered on one of the 225 pixels; any pixel could become the target of the aiming movement. The gaussian distribution was chosen to mirror the fact that the sensory stimulus affected a certain number of

¹The problem has been addressed elsewhere. See for example [21, 25, 27].

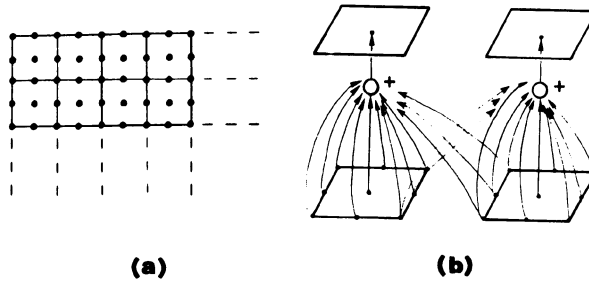


Figure 3: (a) Receptive fields of plan units over the 15x15 array of pixels. (b) The activation of each plan unit is the sum of the values of pixels that belong to its receptive field. Receptive fields are overlapped.

sensory neurons and that the intensity of the stimulus decreased as the distance from the target increased. The values of the different pixels were translated into the activation of plan units by means of coarse coding [15]. Each plan unit had a receptive field over the array of pixels as depicted in Figure 3a; each receptive field contained nine pixels and was overlapped with neighbor fields as shown in Figure 3b. The result was a 7x7 array of plan units. The activation of each plan unit was the sum of the values of the pixels that belonged to its receptive field. The resulting value was then properly normalized. There were many reasons for our preferring coarse coding to local coding (one plan unit for each pixel). For instance, coarse coding made the network learn faster and generalize better because of the overlapping of the activations. In addition, overlapping receptive fields partially overcame the arbitrariness of the workspace discretization by translating a spatial discretization into continuous activations of plan units.

The number of hidden units was empirically fixed at ten. This number resulted from a trade-off between two conflicting needs:

- to provide enough units so that the network would be able to compute the input/output transformation,
- to keep the number of hidden units as low as possible to achieve good generalization. (An excess number of hidden units would make the network act as a look-up table.)

A graphic interface built on top of the simulation software made it relatively easy to investigate the behavior of the hidden layer during the learning phase. We found experimentally that when the number of hidden units was greater than ten learning was not improved and generalization properties were negatively affected.

In summary, the network was composed of:

- 8 output units, one for each muscle of the limb;
- 8 state units, each receiving feedback from one output unit;
- 49 plan units, coarse coding a 15x15 array of pixels;
- 10 hidden units fully connected to plan, state and output units.

All units in the network had continuous activation functions. Output functions of plan units, hidden units and output units were of a logistic type (sigmoids) between a minimum and a maximum value, namely between -1 and 1 for output units and between 0 and 1 for hidden and plan units. Our choice was dictated by the nature of the problem: in this particular implementation (see next section) muscle activations ranged between negative and positive values that were normalized to the interval $[-1,1]$; the stimulus representation ranged between 0 (no stimulus) and a maximum positive value (activation of the unit which contained the gaussian peak), normalized to the interval $[0,1]$. The activation function for state units was linear.

3 Training and Generalization

The network was trained through supervised learning. (See [16] for a review on supervised learning). This class of learning algorithms requires providing a set of input - output pairs. In our case the input was a sensory stimulus and the output was a time trajectory in muscle space. Computing the sequence of muscle activations that correspond to a trajectory of the limb from the initial to the final posture is not straightforward. To this purpose, we used a model which represents a redundant motor system in the form of a network of constraints expressing the geometrical relations among component elements and their steady-state mechanical behavior [33, 34].

This model is based on experimental investigations that stressed the role of muscles' mechanical properties in motor control. These investigations suggest that a muscle is mechanically analogous to a tunable spring. Both muscle and spring are characterized by a set of integrable functions between length and tension at steady state. Neural input (i.e. muscle activation) selects a particular function out of this set [10, 37, 4, 35]. It has been proposed [5, 18] that arm movements are represented and generated by the central nervous system as smooth transitions in posture along virtual trajectories given as time sequences of equilibrium configurations² defined by the muscles' elastic properties.

With a model in which muscles are equivalent to springs, a solution to the problem of motor redundancy is possible. The model makes it possible to compute:

- the xy coordinates of the end-point position of the limb, given the activation of the muscles. (This computation is a well-posed problem.)
- the muscle activations given the xy coordinates of the end-point. (This is an ill-posed problem.)

The problem raised by the redundancy of the motor apparatus is solved by representing the set of muscles as a chain of spring-like elements and by observing that the chain would naturally settle into a configuration of minimum potential energy when perturbed by an external force. The minimum potential-energy criterion makes the solution to the inverse problem unique. The model can then compute a trajectory in muscle space, given the initial and final end-point positions. Also, it is possible to specify the velocity profile along the trajectory (bell-shaped in our case). The result of the computation is a set of eight-dimensional vectors of muscle activations, each vector corresponding to an equilibrium position. Such positions are equi-spaced in time but not in space because of the bell-shaped velocity profile. As far as stiffness is concerned, the model does not

²An equilibrium configuration is defined for a given value of muscle activation as that position at which the forces of opposing muscles generate equal and opposite torques about the joints.

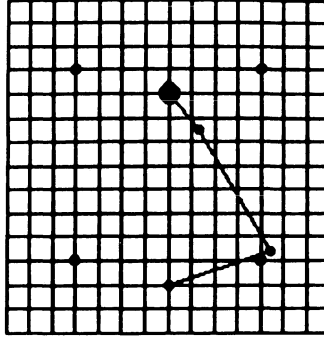


Figure 4: Position of the stimuli for the first four sequences learned by the network.

allow, at this stage of development, direct control over the stiffness values during the transformation from end-point positions to muscle activations. We employed the inverse transformation to compute the output patterns necessary to train the network. The direct transformation (from muscle activation to end-point position) was used during the testing phase.

To train the network we used a standard backpropagation algorithm which makes use of a momentum term; the learning rate was interactively lowered during the training sessions. All trajectories used during the training phase had a duration of six time steps: initial posture, target posture and four intermediate postures. All postures were equilibrium positions as defined by Footnote 2.

One of our major concerns about the training phase was how many and which sequences the network had to learn to correctly generalize the task. We started with four sequences which corresponded to sensory stimuli in the four quadrants into which the limb workspace is ideally divided by the initial end-point position. (See Figure 4.) Figure 5 shows the four trajectories as they were generated by the network after learning. It is worth noting that the bottom-left trajectory contained a joint reversal on the shoulder joint. It took about 4000 trials for the network to learn the four trajectories with the momentum term equal to 0.89 and the learning rate decreasing from 0.2 to 0.01. We used the following convergence criterion: the network was considered to have learned the task when the total square error at the output units was 0.004 on the four sequences. After learning four trajectories, the network was tested for generalization. We did this test by placing stimuli in different positions on the plan units.

Figure 6 shows some of the results. The network produced patterns of muscular activity which made the limb move in the right direction, but there were errors in the generalized trajectories. For example, the end-point did not go correctly to the final position, and the network did not properly adjust the velocity profile. We gradually increased the number of learned trajectories with the purpose of achieving a generalization capability such that the error on the end-point position ³

³Errors were measured, for each end-point position, as the euclidean distance between the end-point position produced by the network and the expected end-point position produced by the mechanical model of muscles.

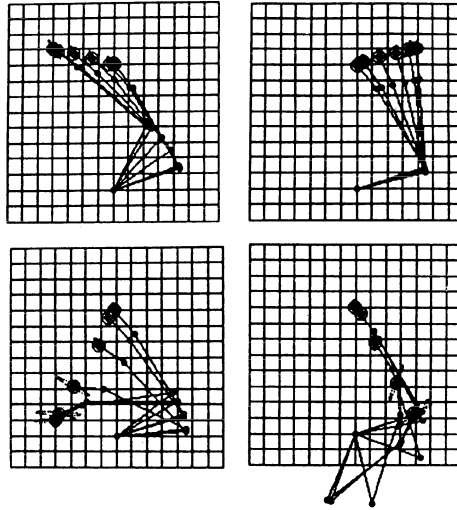


Figure 5: Trajectories towards the stimuli represented in Figure 4. These trajectories are generated by the network after learning. Points along the trajectories are equispaced in time but not in space because of the bell-shaped velocity profile.

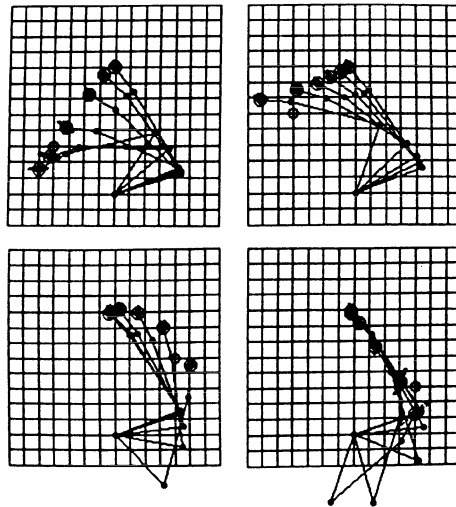


Figure 6: Generalization capability of the network trained with the four sequences represented in Figure 5. The desired final position is circled. The trajectories produced by the network are "wrong," but still the network can produce patterns of muscular activation which make the limb move in the right direction.

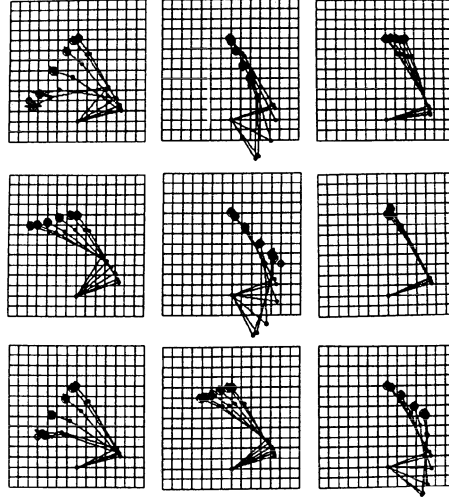


Figure 7: Generalization capability after learning 15 trajectories. The top-left trajectory contains a generalization of the joint reversal on the shoulder. The rightmost trajectory in the second row is a particular case of generalization in which the stimulus was positioned right on the limb end-point. Although the network has not been explicitly taught about the initial posture, it has "understood" how the limb is positioned at the beginning of each trajectory.

for each point on the trajectory did not exceed the grid step. This requirement was equivalent to demanding that the network behave well at the resolution imposed by the discretization of the limb workspace. This level of performance was achieved after the network was taught 15 sequences uniformly distributed over the limb workspace. Figure 7 shows some generalized sequences: here we see that the end-point position is correct along the whole trajectory, and the velocity profile is properly adjusted. In addition, the network generated patterns of muscular activation which corresponded to equilibrium positions of the limb and could produce joint reversals when necessary.

The connectivity pattern produced by learning was organized into inhibitory and excitatory zones. In particular, an analysis of the connections from the hidden units to the output units (interneurons to motoneurons) demonstrated the presence of motor synergies and overlapped functionality in the hidden layer. A detailed description of these results is given in [30].

Three further experiments were performed during the learning phase. First, the learning procedure was repeated by making use of local coding instead of coarse coding (one plan unit for each pixel for a total of 225 plan units). After learning the same 15 sequences, the network was not able to generalize and behaved like a look-up table. In the second experiment, the learning procedure was repeated for a lower resolution on the workspace, which was obtained by doubling the grid step. This doubling led to a 7x7 array of pixels coarse-coded by a 3x3 array of plan units. In this case the network could learn the task (producing errors lower than the grid step) with fewer learned trajectories – 8 as compared to 15. In the third experiment we repeated part of the learning procedure with a linear network obtained simply by removing the hidden layer. The purpose of this experiment was that of investigating the amount of non-linearity present in the input-output transformation. We tried to teach to the linear network the four trajectories shown in Figure 5, first separately and then jointly. We observed the following behavior:

- The linear network could learn the trajectories towards the top-left target and towards the top-right target separately.
- The same two trajectories could not be learned jointly. This fact shows that the linear network could not handle the interferences between the two trajectories, while the non-linear network could.
- The linear network could not learn the trajectories towards the bottom-left target and towards the bottom-right target, neither separately nor jointly.

We concluded that the task is highly non-linear, except in a few peculiar cases. Furthermore, we observed that the trajectories that the linear network could learn were much shorter than those that could not be learnt. Hence, we also investigated the existence of a possible relationship between the task linearity and the length of the trajectories. To this purpose, we tried to teach to the linear network a shorter trajectory in the direction of the bottom right target. The linear network could not learn that trajectory. We concluded that no relationship exists between the trajectory's length and the extent to which the linear approximation holds.

4 Experiments

We performed three experiments with the trained network.

The first experiment was concerned with the duration of the trajectories. Pineda [36] showed that arbitrary networks of logistic units typically have many point attractors. In other words these networks naturally exhibit certain dynamic properties. In our case, the network was instructed during training to produce certain output patterns for six time steps; no instructions were given about what should be done after the sixth time step. We tested the network for 15 time steps, and we observed that in about 80 percent of the cases (i.e. in about 80 percent of the limb workspace), the limb remained steady at the final posture which corresponded to the location of the sensory stimulus. In other words, in 80 percent of the cases, the final posture of the limb acted as a point attractor. In certain parts of the workspace, the limb became unstable after the sixth time step. The part varied with different learning sessions, depending on which solution the network had settled into. There were also cases in which the entire workspace was steady.

The activation of hidden units (i.e. the internal motor image produced by the network for a given trajectory) has been plotted vs. time and hidden units. Figure 8 shows the motor images for four different trajectories; activation of the hidden units was plotted for ten time steps. We observed that: i) in all four cases, the activation patterns at the first time step were the same and corresponded to the initial posture of the limb; ii) the activation remained constant after six time steps and kept the limb steady on the final position.

The second experiment aimed at testing the robustness of the system when the sensory stimulus was varied. The network was trained with stimuli coded as gaussian distributions centered on the target with a certain standard deviation d_0 ; we modified the value of the standard deviation during testing as follows:

$$d_1 = d_0 + 0.1 * d_0$$

$$d_2 = d_0 + 0.2 * d_0$$

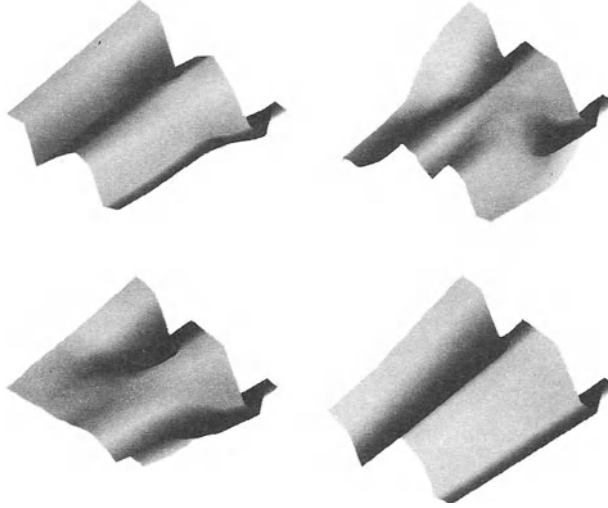


Figure 8: Motor images of four time trajectories. Activation of hidden units is plotted vs. time and hidden units. The activation remains constant after the sixth time step on those values which correspond to the final posture of the limb.

Both of the above cases correspond to a stimulus which is flatter and more spread over the workspace. Figures 9a and 9b show some results of the testing phase in the two cases. We measured the average distance between the trajectories generated with standard deviation d_0 and the trajectories generated with standard deviation d_1 and d_2 . Given the stimulus d_1 the average distance was lower than 0.4; in the second case the average distance was higher (around 0.7), which resulted in trajectories that were "noisy", but still acceptable. This experiment showed that the architecture was reasonably robust in the face of slight changes in the stimulus representation.

The third experiment was performed with the aid of a double target. A sensory stimulus was given to the network. As the arm was moving in steps toward the stimulus, the stimulus was turned off, and another stimulus at a different location was turned on. When this occurred, the arm switched direction towards the new target. The experiment was repeated in the following two cases:

1. with the duration of first stimulus corresponding to the first two time steps made by the limb;
2. with the duration of first stimulus corresponding to the first three time steps.

Figure 10 shows the resulting trajectories for the first case, while Figure 11 represents the trajectories for the second case. Note that in both cases the limb reached the second target. The results of this experiment show that the network, having learned how to reach a set of targets from a fixed initial position of the limb, was also able to reach the same set of targets from a different posture, the one in which the limb was positioned when the second target was turned on. This result indicates that the network was able to solve the so-called motor-equivalence problem. Experiments on intact biological systems [12, 28] clearly show that the shape and length of the trajectories generated by a double target experiment depend upon the duration of the first stimulus. The network in our experiments seemed to be insensitive to this parameter (compare Figures

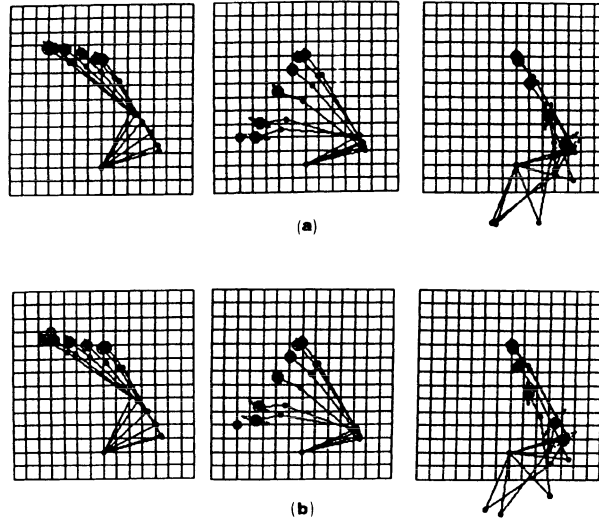


Figure 9: (a) Trajectories generated by the network when the stimulus is coded as a gaussian distribution with standard deviation $d_1 = d_0 + 0.1 * d_0$. (b) Trajectories generated by the network when the stimulus is coded by a gaussian distribution with standard deviation $d_2 = d_0 + 0.2 * d_0$. In this case, the trajectories look "noisy," but the network can still solve the motor problem.

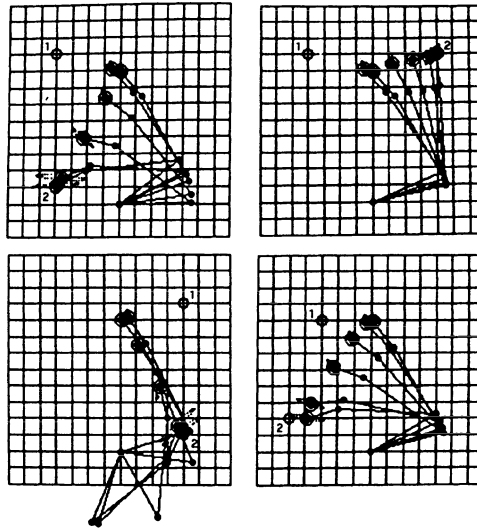


Figure 10: Double target experiment. The first target was turned on for two time steps, and then it was turned off. The experiment was performed on both learned and generalized trajectories.

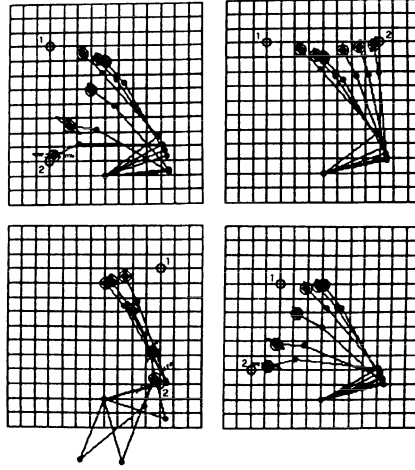


Figure 11: Double target experiment. The first target was turned on for three time steps, and then it was turned off. The experiment was performed on both learned and generalized trajectories.

10 and 11). In a sense, the network "forgot" everything about the first trajectory when the second stimulus was turned on. These different behaviors may indicate that a pure executorial system - like the neural network described in this paper - does not contain any smoothing mechanism, whereas such mechanisms are present when planning occurs. In other words, trajectory smoothing is not a direct consequence of the mechanical properties of the actuator, but the result of some specialized brain functions. With neural networks, one could obtain a smoothing behavior by enriching the network with other blocks which somehow implement the above mentioned brain functions. Alternatively, one could add some dynamics (i.e. self-connections) at the output units (like in Jordan's flow networks [22]) to make the next output of the network a function of the past outputs. The latter solution would correspond to assuming that trajectory smoothing is a low-level operation handled by motoneurons.

5 Conclusions

In this paper we presented a model for the formation of limb trajectories, based on a neural network architecture. The task under consideration was that of reaching a target defined in terms of a sensory stimuli. The trajectories had a bell-shaped velocity profile. The network produced trajectories in muscle space, which were translated into end-point space by means of a model which takes into account the elastic properties of muscles [34]. The same model also made it possible to compute the inverse transformation - from end point space to muscle space. That transformation was used to generate the training sequences. The particular architecture used for producing time trajectories was that proposed by Jordan [20].

The results of experiments performed during and after learning show that:

1. The task can be learned by a three-layer sequential network trained by a standard back-propagation procedure.

2. The network successfully generalizes in trajectory space: the error of the generalized trajectories measured at each time step can be made lower than the discretization step of the limb workspace. Moreover, the velocity profiles generalizes appropriately.
3. The same task cannot be learned by a linear network.
4. The model is robust with respect to the input signals: slight changes to the stimulus coding do not significantly affect the network overall performance.
5. The network spontaneously exhibit attractor-dynamics properties. Final end-point positions behave like point attractors in the majority of the limb workspace.
6. The network can solve the motor-equivalence problem as shown by the double-target experiment. The network does not exhibit smoothing properties, and seems to be insensitive to the duration of the first stimulus.

In our model the information about the actual position of the end-point was not explicitly computed. (It is only implicitly available through the muscles' model.) This fact did not represent a limitation for the task under consideration, since the task was performed, as already pointed out, at the reflex level, without any planning. However, in human experiments Morasso [32] showed that the information about the end-point position plays a crucial role at the planning level. If planning were to be incorporated in our model, its architecture ought to be expanded to include explicit computation of the end-point position.

As far as task representation is concerned, we merged the kinematic problem and the velocity profile into a single three-layer network, but this was not the only possible choice; the two problems could as well have been separately addressed and represented by means of two interconnected networks. The latter possibility has been investigated by Jordan [23].

The work described here has relevance to the robotics research since it may suggest some basic principles for designing artificial limbs whose structure is inspired by natural systems [7].

As far as neuroscience is concerned, the relationship between the research covered in this paper and the organization of biological systems is an open problem and will be the object of further investigations.

Current research activities are concerned with i) a deeper investigation of the motor-equivalence task by extending learning to multiple starting postures, ii) the study of motor synergies in the hidden layer.

6 References

1. Abend W, Bizzi E, Morasso P (1982) Human Arm Trajectory Formation, *Brain*, 105, pp. 331-348.
2. Atkeson CG, Hollerbach JM (1985) Kinematic Features of Unrestrained Arm Movements, *J. of Neurosci.*, 5(9), pp. 2318-2330.
3. Berkinblitt MB, Feldman AG, Fulkson OI (1986) Adaptability of Innate Motor Patterns and Motor Control Mechanisms, *Behavioral and Brain Sci.*, 9, pp. 585-638.

4. Bizzi E, Polit A, Morasso P (1976) Mechanisms Underlying Achievement of Final Head Position, *J. of Neurophysiology*, 39, 435.
5. Bizzi E, Accornero N, Chaple W, Hogan N (1984) Posture Control and Trajectory Formation during Arm Movement, *Journal of Neuroscience*, 4(11), 2738-2744.
6. Bullock D, Grossberg S (1988) Neural Dynamics of Planned Arm Movements: Emergent Invariants and Speed Accuracy Properties During Trajectory Formation, in *Neural Networks and Natural Intelligence*, S. Grossberg Ed., pp. 553-622, MIT Press.
7. De Rossi D, Casalino G, Morasso P (1988) Skin and Muscles: the Basic Wetware for Sensory-Motor Control, 1st Annual INNS Meeting, Boston (MA), pp.332.
8. Eckmiller R (1988) Concept of a 4-Joint Machine with Neural Network Control for the Generation of Two-Dimensional Trajectories, 1st Annual INNS Meeting, Boston (MA), pp 334.
9. Edelman GM (1987) Neural Darwinism - The Theory of Neuronal Group Selection, Basic Books Inc.
10. Feldman AG (1986) Once More on the Equilibrium Point Hypothesis (λ model) for Motor Control, *Journal of Motor Behavior*, 18, pp. 17-54.
11. Flash T, Hogan N (1985) The Coordination of Arm Movements: an Experimentally Confirmed Mathematical Model, *Journal of Neuroscience*, 5(7), pp. 1688-1703.
12. Georgopoulos A, Kalaska JF, Massey JT (1981) Spatial Trajectories and Reaction Times of Aimed Movements: Effects of Practice, Uncertainty and Change in Target Location, *Journal of Neurophysiology*, 46(4), pp. 725-743.
13. Georgopoulos A (1986) On Reaching, *Annual Review of Neuroscience*, 9, pp. 147-170.
14. Giszter S, McIntyre J, Bizzi E (1989) Kinematic Strategies and Sensorimotor Transformations in the Wiping Movements of Frogs, *J. of Neurophysiology*, 62(3), pp. 750-767.
15. Hinton GE, McClelland JL, Rumelhart DE (1986) Distributed Representations, in *Parallel Distributed Processing*, McClelland JL, Rumelhart DE Eds., MIT Press, Cambridge (MA).
16. Hinton GE (1987) Connectionist Learning Procedures, Tech. Rep. CMU-CS-87-115.
17. Hogan N (1984) Some Computational Problems Simplified by Impedance Control, *Proc. ASME Computers in Engineering*, Las Vegas.
18. Hogan N (1985) The Mechanic of Multi-Joint Posture and Movement Control, *Biol. Cybern.*, 53 (1).
19. Howarth CI, Beggs WDA (1981) Discrete Movements, in D. Holding Ed., *Human Skills*, New York:Wiley and Sons, pp. 91-117.
20. Jordan MI (1986) Attractor Dynamics and Parallelism in a Connectionist Sequential Machine, *Proc. 8th Annual Conf. of the Cognitive Science Society*, Hillsdale, N.J.: Erlbaum.
21. Jordan MI (1988) Supervised Learning and Systems with Excess Degrees of Freedom, COINS Tech. Rep., Univ. of Massachusetts at Amherst.

22. Jordan MI (1989a) Serial Order: a Parallel Distributed Processing Approach, in *Connectionism and Speech*, J.L. Elman and D.E. Rumelhart Eds., Hillsdale, NJ: Erlbaum, in press.
23. Jordan MI (1989b) Indeterminate Motor Skill Learning Problems, in *Attention and Performance XIII*, M. Jeannerod Ed., Hillsdale NJ: Erlbaum.
24. Kawato M, Furukawa M, Suzuki R (1987) A Hierarchical Neural-Network Model for Control and Learning of Voluntary Movements, *Biol. Cybern.* 57, pp. 169-185.
25. Kawato M, Isobe M, Maeda Y, Suzuki R (1988) Coordinates Transformation and Learning Control for Visually Guided Voluntary Movements with Iteration: a Newton-like Method in Function Space, *Biol. Cybern.*, 59, pp. 161-177.
26. Kelso JAS (1982) *Human Motor Behavior*, Hillsdale NJ: Erlbaum.
27. Kuperstein M (1988) Neural Model of Adaptive Hand-Eye Coordination for Single Postures, *Science*, Vol. 239, pp. 1308-1311.
28. Massey JT, Schwartz AB, Georgopoulos A (1986) On Information Processing and Performing a Movement Sequence, in C. Fraum and H. Henvner Eds. *Generation and Modulation of Action Patterns*, Experimental Brain Research Supplement.
29. Massone L, Bizzi E (1989a) Generation of Limb Trajectories with a Sequential Network, *Proc. Int. Joint Conf. on Neural Networks*, June 18-22, Washington D.C.
30. Massone L, Bizzi E (1989b) A Neural Network Model for Limb Trajectory Formation, *Biol. Cybern.*, 61, pp. 417-425.
31. Miyamoto H, Kawato M, Setoyama T, Suzuki R (1988) Feedback Error Learning Neural Network for Trajectory Control of Robotic Manipulator, *Neural Networks*, vol. 1, pp. 251-265.
32. Morasso P (1981) Spatial Control of Arm Movements, *Experimental Brain Research*, 42, pp. 223-227.
33. Mussa Ivaldi FA, McIntyre J, Bizzi E (1988a) Theoretical and Experimental Perspectives on Arm Trajectory Formation: A Distributed Model for Motor Redundancy, in *Biological and Artificial Intelligence Systems*, E. Clementi and S. Chin Eds., pp. 563-577, Escom.
34. Mussa Ivaldi FA, Morasso P, Zaccaria R (1988b) Kinematic Networks - A Distributed Model for Representing and Regularizing Motor Redundancy, *Biol. Cybern.*, 60, pp. 1-16.
35. Nichols TR, Houk JC (1976) Improvement in Linearity and Regulation of Stiffness that Results from Action of Stretch Reflex, *J. Neurophysiol.*, 29 (119).
36. Pineda FJ (1987) Generalization of Backpropagation to Recurrent and Higher Order Neural Networks, *Proc. IEEE Conf. on Neural Networks and Information Processing Systems*, Denver (CO).
37. Rack PMH, Westbury DR (1969) The Effects of Length and Stimulus Rate on Tension in Isometric Cat Soleus Muscle, *J. Physiol.*, 204(443).
38. Schoner G, Kelso JAS (1988) Dynamic Pattern Generation in Behavioral and Neural Systems, *Science*, 239, pp. 1513-1520.

39. Shadmer R, Lindquist GD (1988) A Neural Model for Modifiable Pattern Generation in the Scratch Reflex, 1st INNS Meeting, Boston (MA), pp. 359.]

ACKNOWLEDGEMENTS

The authors wish to thank Michael Jordan for his constant support and valuable suggestions, and for making available the basic software which implements sequential networks. Thanks also go to Simon Giszter, Joe McIntyre, Ferdinando Mussa Ivaldi and Tomaso Poggio. This work was supported by the Office of Naval Research Grant N00014/88/k/0372, the Sloan Foundation and a NATO Advanced Research Fellowship Program.

Part 5

Design Technologies

Flexible Robot Manipulators and Grippers: Relatives of Elephant Trunks and Squid Tentacles

J.F. Wilson¹, D. Li², Z. Chen³, and R.T. George, Jr.¹

¹School of Engineering, Duke University, Durham, N.C. 27706, U.S.A.

²Tsinghua University, Beijing, China

³Department of Electrical Engineering, North Carolina State University, Raleigh, N.C. 27695

Abstract. Elephants and squids have continuously flexible appendages that are well adapted for manipulating and gripping. In this paper we briefly review the overall structural forms and motions of elephant trunks and squid tentacles. We then discuss how we incorporated some of their biological characteristics in the design of a flexible arm manipulator with open loop control (Part 1) and in the design of a flexible, two-fingered gripper with closed loop control (Part 2).

Introduction

Both elephant trunks and squid tentacles are hydrostatic structures; that is, these appendages maintain constant volume as selected muscle groups surrounding this volume contract to achieve particular motions. Cross sections of a trunk and tentacle are depicted in Fig. 1a and Fig. 1b. In both structures, uniform contraction of the longitudinal muscles effects uniform shortening, contraction of longitudinal muscles along one side effects bending, contraction of both circumferential and radial muscles effects elongation, and contraction of the right or left-handed helical muscles leads to respective rotations of the appendage about the longitudinal axis. For detailed discussions of elephant trunk morphology see Wilson et al (1989), and for squid tentacle morphology see Kier and Smith (1985).

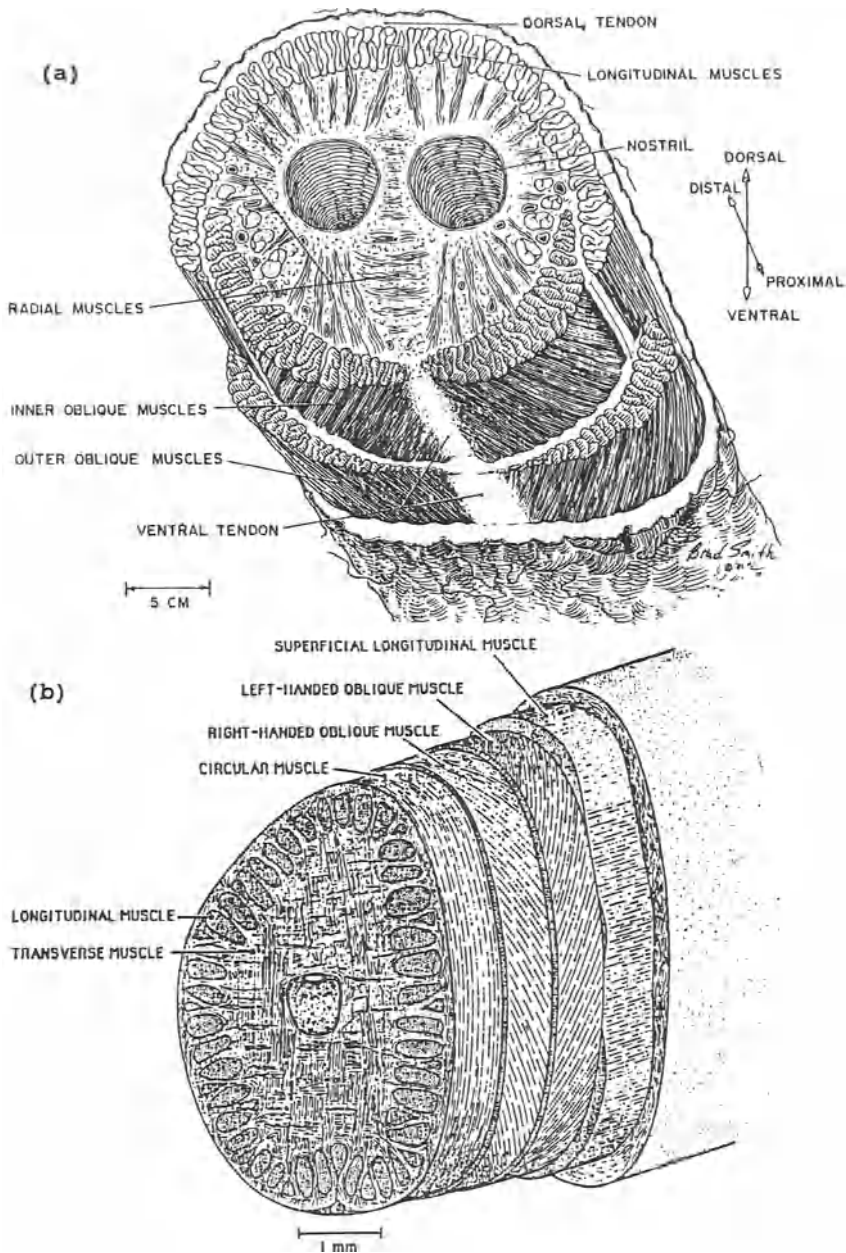
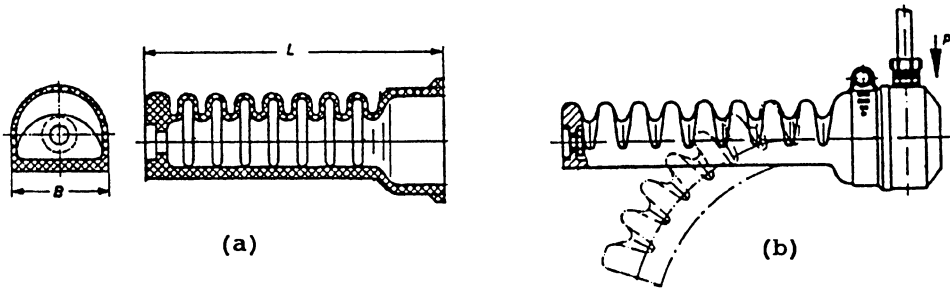


Fig. 1 Cross sections of the muscle structure for:
 (a) Trunk of a mature female Asian elephant (drawing by Brad Smith);
 (b) Tentacle of a squid, *Loligo pealei* (adapted from Kier and Smith, 1985).



ELEMENT TYPE	DIMENSIONS (mm)	
	B	L
A-1, A-2	40	130
B-1, B-2	28	92
C-1, C-2	20	65

Fig. 2 Polyurethane elements (Simrit Corp., Arlington Heights, IL): (a) unpressurized, (b) pressurized.

Orthotropic, polyurethane tubes that bend when pressurized are depicted in Fig. 2. Placed end-to-end, such elements may simulate the actions of the elephant trunk and the squid tentacle, as discussed by Palaniappan (1985), Ghattas (1988), Wilson and Snyder (1988), Wilson and Mahajan (1989), and Snyder and Wilson (1989). In addition, end effectors for such arms may be configured of such flexible tubes in parallel to form gripping fingers. Although there appears to be no published results on such flexible, pneumatically driven fingers, there are many excellent articles on the mechanics and control of rigid-linked grippers: the especially thorough reviews of the literature by Dario and Buttazzo (1987), Iberall (1987), Li and Sastry (1987), and Nguyen (1989); the recent research of Backé (1986), Kerr and Roth (1986), Kumar and Waldron (1987), and Lee (1988); and the earlier works of Cutkowsky (1984), Datseris and Palm (1984), Hanafusa and Asada (1977), Jacobson et al (1984), Salisbury and Craig (1982), and Wilson (1984a, 1984b).

In the present simulations of arm and finger action, we use the general type of commercially available tube element depicted in Fig. 2a and the three element geometries A, B, C whose overall dimensions are given in the table insert of that figure. These tube elements have a convoluted bellows section for part of the circumference and a flat side that contains its neutral axis of bending. Since the center of pressure on the rigid end caps is offset from the neutral axis of bending, cantilevered elements bend as shown in Fig. 2b when subjected to internal pressure.

In the design of pressure tube systems for robotic applications, it is important to know the mechanical behavior of an individual element under various loading conditions. For the tube element shown in Fig. 2, the measured results are shown in Figs. 3-5. The type of experiment and its variables such as loading

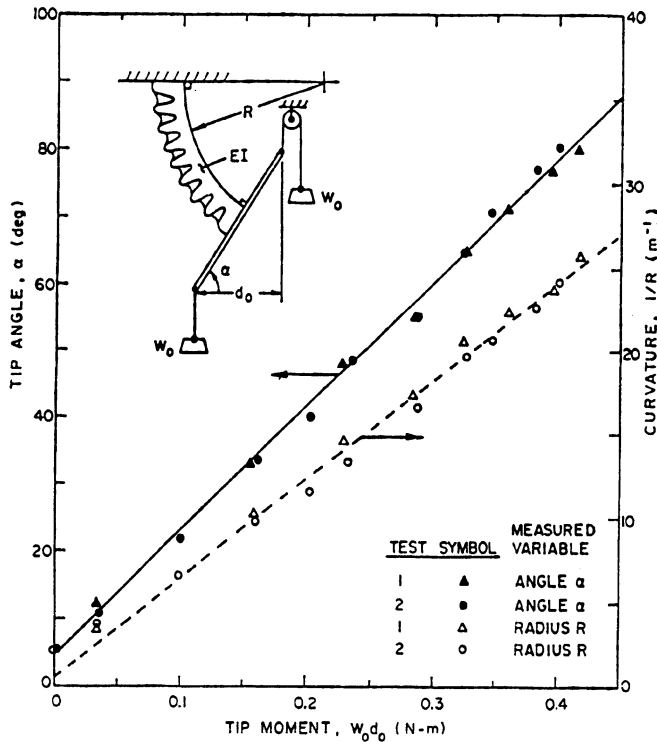


Fig. 3 Elastic behavior of polyurethane element B of Fig. 2, in response to a moment applied to the tip; (from Snyder and Wilson, 1990, and reprinted with permission of ASME).

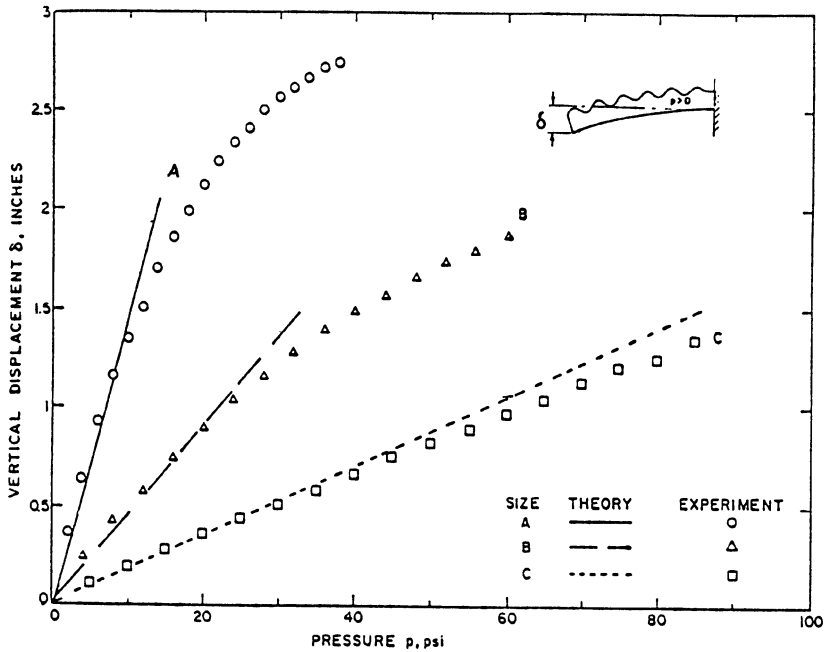


Fig. 4 Transverse tip displacement of polyurethane elements A, B and C under internal pressure only.

and deflection are defined in the inserts of these figures. Thus Fig. 3 shows, for element type B, that the tip angle α and its curvature $1/R$ both vary linearly with the applied uniform moment at its tip. Figure 4 shows that elements without a transverse tip load have a linear range of transverse tip deflection with internal pressure. Figure 5 shows the transverse tip load F required to suppress the transverse tip deflection.

Our main objective in this paper is to show through experiments that combinations of such tubes, with proper pressure controls, may be used in applications involving manipulation. In Part I we discuss experiments on a simulated elephant trunk lifting a payload: a simulation comprised of several tubes combined in series, with open loop pressure control, and used in pick-and-place scenarios. In Part II we discuss experiments on a simulated pair of squid tentacles: a simulation comprised of two tubes combined in parallel, with closed loop pressure control, and used to achieve a constant gripping force on an object being manipulated within its grasp.

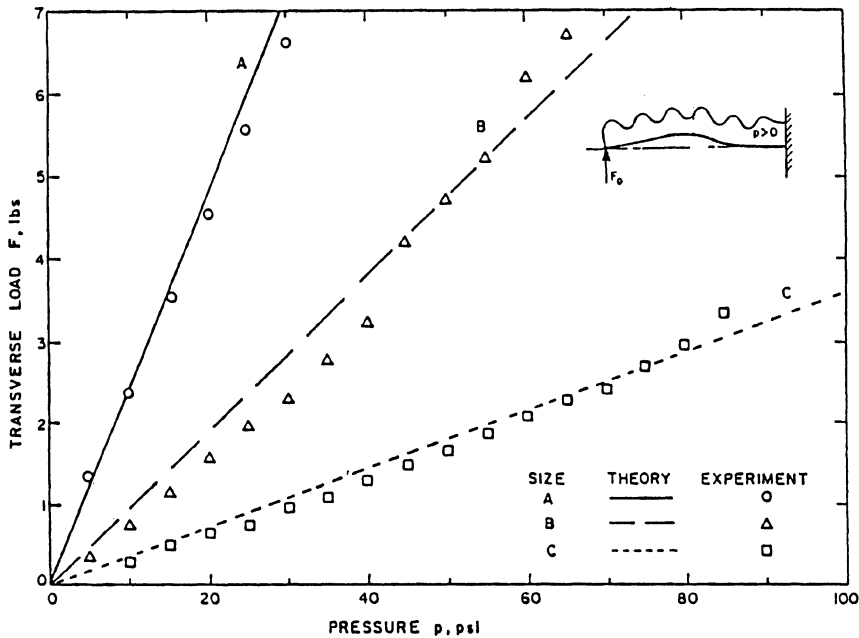


Fig. 5 Transverse tip load required to suppress tip displacement for pressurized, polyurethane elements A, B, and C.

1. Flexible Robotic Arm Manipulator

The counterpart of the elephant trunk, or the arm manipulator and its end effector (gripper), are shown in Fig. 6. This system, discussed in detail by Wilson (1987), consists of software Programs I and II used for open loop control of the arm and gripper. The basic system design is outlined now.

Mechanical Design

The arm manipulator, Fig. 6, is designed from the three different sizes of polyurethane elements listed in the table insert of Fig. 2. The letters A, B, C designate the size and the numbers after these letters designate the element position in the arm. This arm resembles somewhat the trunk of an elephant in both its out-

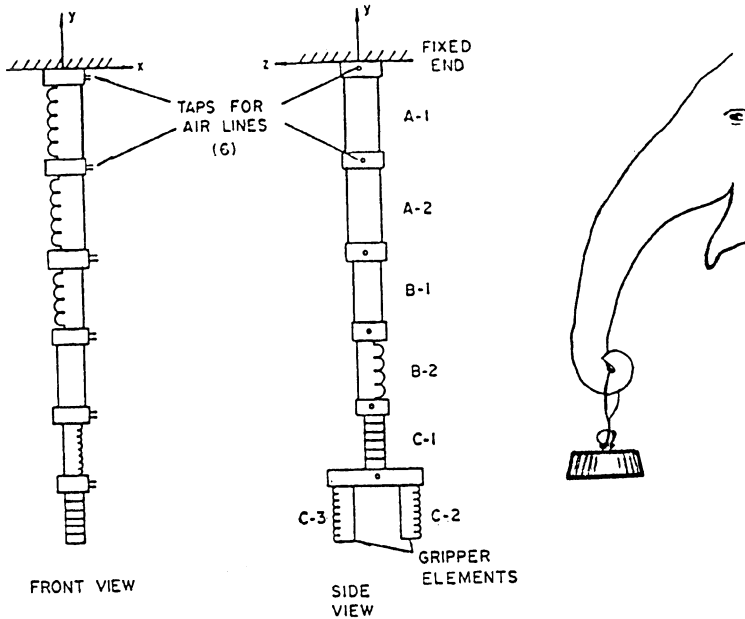


Fig. 6 Flexible arm and gripper, counterpart to the elephant trunk.

ward form and manipulative function and hangs vertically when unpressurized and at rest. The largest elements A-1 and A-2 are at the base and sustain the highest arm moments when the pressurized arm with the payload at the gripper is displaced from the vertical equilibrium position. In terms of the Cartesian coordinate system defined in Fig. 6, arm elements A-1, A-2, B-1, and C-1 are aligned to bend in the x,y -plane; but element B-2 is aligned to bend in the y,z -plane. The latter element is effective in avoiding collisions of the arm with obstacles in the x,y -plane. As discussed below, each arm element is a distinct pressure cell and has its own air supply line connected to the control module. The two opposing elements of the gripper, however, have a common air supply.

Hardware Control System

Shown in Fig. 7a and 7b, respectively, are the fluid circuit and electronic circuit for a single tube element. Each fluid circuit

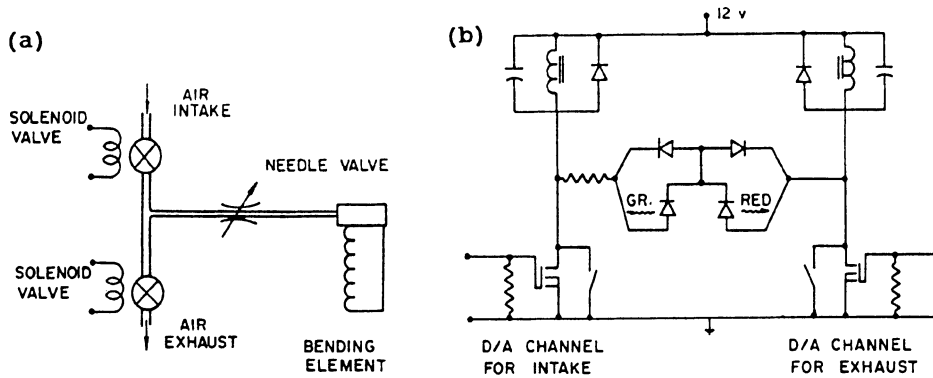


Fig. 7 Circuits for open loop control of a single element: (a) Fluid circuit, (b) Electronic circuit for D/A control of air intake, exhaust.

has two solenoid valves, one for air admission and one for air exhaust, and a hand adjustable needle valve to slow the flow of air, if necessary. Each electronic circuit has two digital channels, one for air admission and one for air exhaust. The system control module consists of six dual units of Fig. 7, one unit for each of the five arm elements and one for the gripper. For arm elements, the admitted air is provided from a 275 kPa regulated source; and for the gripper, air is provided from a 480 kPa regulated source to assure adequate force to grip the payload.

The interface between the system control module and the microcomputer was accomplished with logic-level MOSFET (metal-oxide-semiconductor field-effect transistor) power switches capable of passing the current required by the solenoid valves and of withstanding the voltage surges at solenoid turn-off. Each solenoid was provided with a diode and a capacitor to limit the voltage surge. For control, logic-level MOSFETs accepted signal voltages available from the computer. Logic 0 (0 volts) produced a switch-open effect, and logic 1 (3.5 to 5 volts) produced a switch-closed effect. The interface was designed to operate with a microcomputer through a 8255 parallel output device which was a Techmar Lab Tender board in an IBM-pc-XT.

Two Software Control Methods

Tests of the solenoid valves and the individual bending elements defined in Fig. 2 showed sufficiently small response times for achieving reasonably fast action of the manipulator arm and gripper: 12 msec for the solenoids, 1 sec for the largest bending element A to achieve a 90 deg bend, and 0.3 sec response time for elements B and C.

Tempered with a knowledge of these response times, we developed two open-loop software control methods to achieve pick-and-place arm maneuvers. The first method employs Program I, a timed sequencing method involving incrementing or decrementing the air pressure in each element. For example, Fig. 8 depicts the time sequences during which the intake and exhaust solenoid valves are open to drive the bending elements. The mathematical description of valve operation is a sequence of step functions to open, each followed after some time by a step to close.

The second method employs Program II, or a pressure pulsing method. Figure 9 depicts the control signal for an intake or exhaust solenoid valve in which the valve is open during the pulse width time T_1 and closed during the pulse gap time T_2 . Here, the desired arm and gripper motion is achieved by selecting the pulse width, the pulse gap, and the number of pulses of each solenoid.

Experiments and Discussion of Results

A typical pick-and-place maneuver was chosen to evaluate arm performance, and especially to determine the shortest possible cycle times consistent with both smooth arm motion and a minimum overshoot of the target point and final rest point. Stroboscopic views of an arm and gripper for a typical experiment are shown in Fig. 10. Here, the gripper is pressurized and grasps an object from the lower platform; the arm elements are selectively pressurized to lift the object to an upper platform while avoiding a collision with it; the gripper is depressurized and the object is released; the arm elements are repressurized to raise the arm somewhat; and then all elements are depressurized as the manipulator returns to its initial vertical equilibrium position.

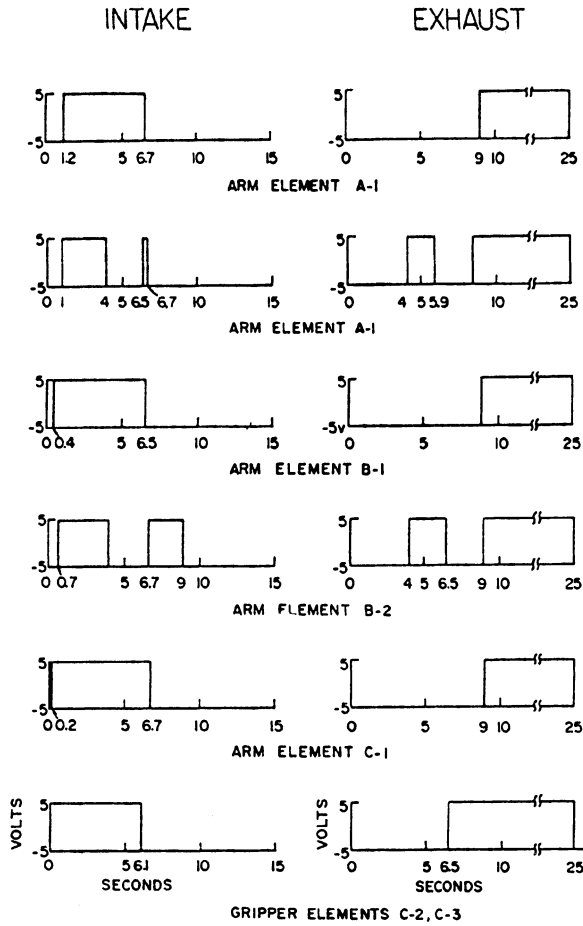


Fig. 8 Timed sequencing for a typical arm maneuver.

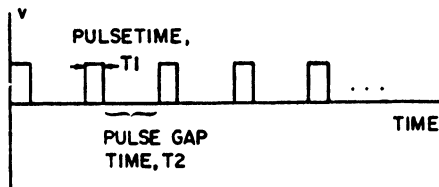


Fig. 9 Definition of the control signal for the pulsing method.

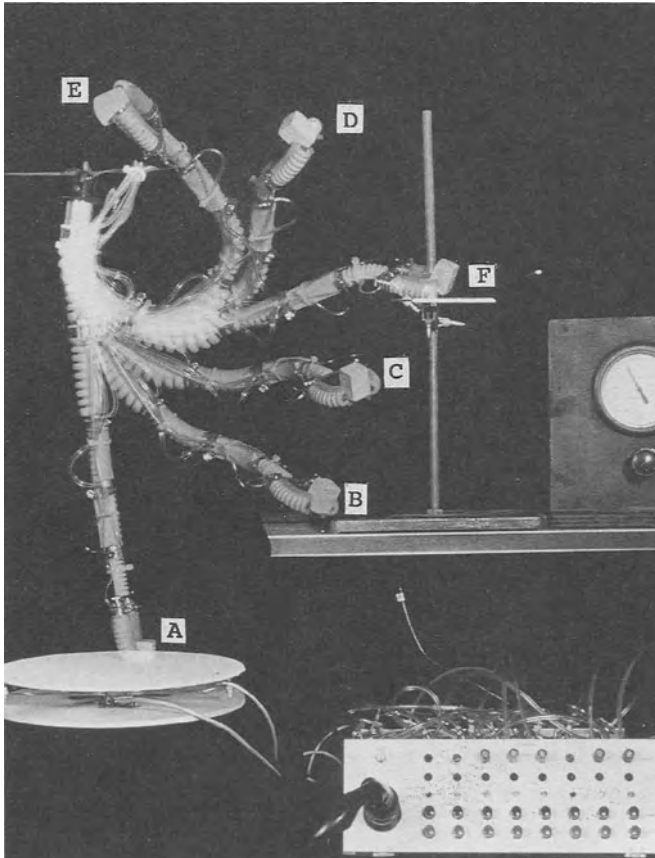


Fig. 10 Stroboscopic photographs for a pick-and-place maneuver: A is the pickup point; B, C, D, and E follow; and F is the placement point for the block on the upper platform. (Photograph by Anthony Benson, courtesy of Scientific American)

The coordinate system of the arm's work space for our series of experiments is defined in Fig. 11. In all experiments, our maneuvered object was a wood cube with an edge dimension of 2.54 cm, initially oriented with its sides parallel to the coordinate planes, with its cg located at position $X = Y = Z = 0$. After the gripper grasped the cube from this location on the lower platform, the arm deposited the cube at the target point on the upper platform, position $X = 47$ cm, $Y = 49$ cm, and $Z = 5$ cm, and the arm returned to its initial vertical position. The total time

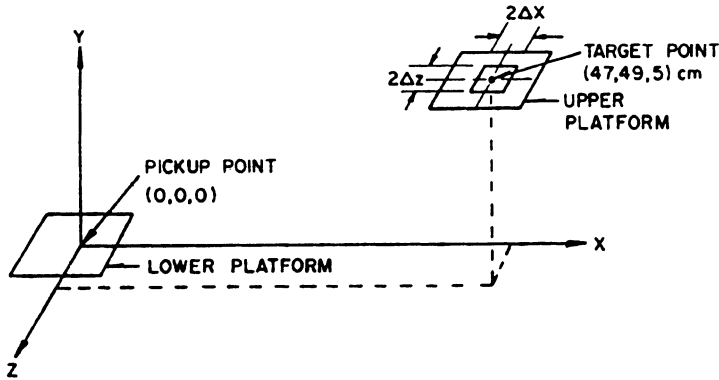


Fig. 11 Coordinate system for the pick-and-place maneuver.

for this maneuver is defined as the cycle time T .

We evaluated the performance of the manipulator and the two control methods several ways: qualitative observations of the smoothness of arm motion; quantitative data on the amplitude and frequency of arm motion at the end of the maneuver; measurements of the deviations ΔX , ΔY , ΔZ from their respective target points; and the change in rotation $\Delta\theta$ of the cube about the Y-axis, from before to after placement.

In the first series of experiments, we employed Program I, the timed sequencing control method with the pressure patterns of Fig 8. We needed to perform several experiments to determine the needle valve settings that minimized unwanted arm oscillations. Qualitative observations of these experiments are summarized as follows. The smoothest arm motion occurred when the needle valves were set to greatly restrict flow of air and where the maneuver cycle times T were 25 seconds. As the needle valves were opened to allow for higher air flow rates, the arm moved faster, and values of T as low as 5 seconds could be achieved. However, we achieved this fast action only at the expense of increased arm oscillations during and immediately after the maneuver cycle.

By contrast, we found that Program II gave superior performance. When we set the needle valves to the fully open position and employed the pulsing control method of Fig. 9, the arm action

was as smooth at cycle times of 5 sec as for the timed sequencing control method at a 25 sec cycle time.

Typical experimental results showing the accuracy of object placement for the pick-and-place maneuver are summarized in Table 1. For both Programs I and II, the time intervals for the maneuver cycle, the pulse, and the gap were fixed. For each program, the change in the object rotation θ and the deviations X and Z from the object's target values were measured in repeated trials. Table 1 gives the maximum measured θ and the maximum percent deviation from the target coordinates, based on 10 trials for each program. The right-most column lists the maximum amplitudes A of free oscillations that occur at the end of the cycles where the arm is depressurized and then swings about its vertical equilibrium position.

The experimental results for the two methods of arm motion control are compared and summarized as follows.

1. If the gap time interval T_2 is chosen as zero for pulsing control, Program II of Table 1, then the two control methods are the same.

2. Both control methods may be used successfully for pick-and-place manipulation cycle times T as small as 4.2 seconds.

3. Pulsing is superior to timed sequencing because the pulsing method affords more versatile programming and gives smoother arm motion at a fixed T .

4. The object positioning errors for X and Z were at most 9% for $T_2 = 0$ and $T = 4.2$ sec. However, these errors decreased when both T_2 and T were increased.

5. Object rotations $\Delta\theta$ up to 22 deg were performed.

6. The amplitude A of the end-cycle, free swinging oscillations were at most 7 cm, but this value decreased when both T_2 and T were increased.

An increase in object placement accuracy and a decrease in end cycle oscillations may be effected by introducing positioning feedback and damping in the control system. However, the present new-concept manipulator arm with its pulsing, open-loop control method may be used in applications where high accuracy of object placement is not a requirement; and where fast-acting, lightweight, robust, and relatively simple arm structures are needed.

2. Flexible Gripper

Shown in Fig. 12 are a two-fingered gripper and a squid whose tentacles are grasping a shrimp. There is a resemblance in the motion of the experimental gripper and that of the tentacles. The two fingers of our gripper were type B polyurethane tube elements of Fig. 2. With the aid of a force sensor at the base of the gripper, a pressure module, a microcomputer, and interface equipment, we implemented the closed-loop control system (two types) so that the fingers could maintain the gripping force even as arbitrary side loads were applied to one of the fingers. The essential features of this design are summarized as follows.

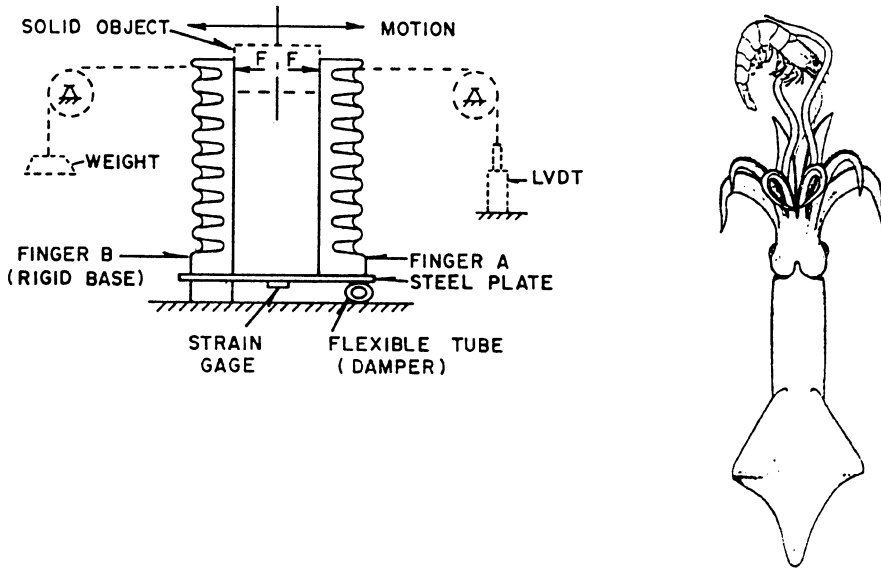


Fig. 12 Mechanical design of the controlled gripper made of two polyurethane elements B, counterpart to a pair of squid tentacles gripping a shrimp. (Squid picture adapted from a photograph by Kier and Smith, 1985).

Mechanical Design

As shown in Fig. 12, a finger is mounted at each end of a relatively stiff plate of spring steel. The transverse gripping force F at a fixed position L on each finger produces a uniform bending moment of magnitude $L \cdot F$ in the plate and a corresponding strain ϵ that varies linearly with F . The soft plastic tube under the plate at finger A serves as a hinge support, provides damping, and effects a sufficiently high natural frequency of 25 hz for this mechanical assembly, well above the 6 hz resonance measured for the electronic circuit/pneumatic valve system described presently.

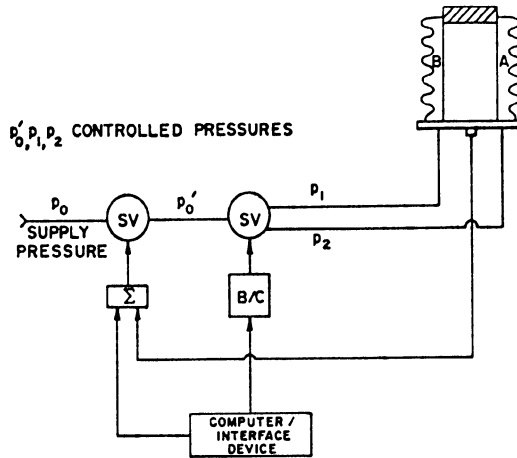
As each finger is pressurized, it bends inward to clamp the solid object at a predetermined gripping force F and the plate strain ϵ is measured. The strain sensor is a foil-type strain gage that forms one arm of a Wheatstone bridge circuit, where the unbalance of the bridge voltage, measured through an appropriate signal conditioner, gives a measure of ϵ . As the fingers manipulate the gripped object in a prescribed way, the air pressure in each finger is regulated so that ϵ remains constant, or nearly so. In this way, the fingers' gripping force remains essentially constant even when external side loads are applied, as shown by the broken lines of Fig. 12. The linear variable differential transformer (LVDT) shown in this figure is used to measure the horizontal displacement of the fingers, but is not a part of the finger manipulation and control system.

Control System and Computer Interface

Two control systems were developed. Control Method 1, shown in Fig. 13a, provides force control through the first servovalve which sets the internal air pressure. Relative finger movements are controlled by the second servovalve, the positioning valve. Control Method 2, shown in Fig. 13b, employs one servovalve per finger. Finger A movement is set by computer control and Finger B is moved under computer control to maintain the desired clamping force. This is done through feedback from the strain gage.

Each control system provides accurate control of the gripping force and the horizontal finger position. Each system uses

(a)



B/C BUFFER / CURRENT DRIVE CIRCUIT

Σ CONTROL CIRCUIT

SV SERVOVALVE

(b)

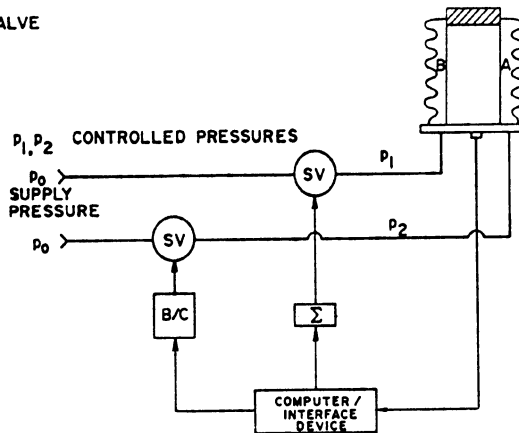


Fig. 13 Control systems for two-fingered gripper experiments:
 (a) Control Method 1, (b) Control Method 2. Servo-
 valves: No. 200 PN, Atchley Controls, Los Angeles, CA.

a microcomputer for finger control and a monitor displays the dynamic gripping force. Analog inputs for the feedback controlled gripper force are the strain gage voltage and the reference voltage proportional to the preset gripper force F . The reference voltage is obtained from an Apple II computer output port.

An interface, an ISAAC 91A, was added to the Apple II computer to handle the analog input and output signals. Differential buffer circuits follow the interface outputs for signal gain and thereby increased the capability for sourcing/sinking current of + 20 mA. Strain gage sensing provides a negative feedback voltage to maintain the desired gripping force.

The system response time is limited by its mechanical resonance frequency of 25 Hz. An increase in the feedback system gain decreases the response time and increases overshoot. Practical settings are: 0 to 5 V for strain gage input; -5 to +5 V for the computer/interface outputs; and a buffer gain of 33 to allow for quick response time without exciting resonance behavior. Undesirably large signals are controlled by Zener diodes in the buffer circuit. Details of the circuit designs and printouts of the the control software are given by Wilson (1984b).

Experiments and Discussion of Results

The two-fingered gripper was evaluated in a series of experiments, eight of which are presented herein. The purposes of these experiments were: to compare the response times and accuracy of the two control systems; to measure reproducibility of finger displacement; and to measure how well the fingers could maintain a constant gripping force on an object, both with and without horizontal side loads applied to each finger.

In each of these eight experiments, the fingers were programmed to grip a block of wood of dimensions 2.54 by 5 by 5 cm with a preset force F of 8.1 N, and to move this block side to side with nearly harmonic displacement amplitudes of + 1.9 cm. Listed on the left side of Table 2 are the sets of different conditions imposed for each experiment: the cycle times of 1.6 or 2 sec, and the horizontal side loads (see Fig 12) of 0, 1.42 N, or 0.86 N.

The experimental results for the two methods of gripping are summarized and compared as follows. See the right side of Table 2 for measured values of gripping force.

1. In the absence of horizontal side loads on Fingers A and B, there was very little difference in the accuracy with which

these two control systems maintained the prescribed value of F : within 1.8% for Method 1 and within 2% for Method 2. This was true for both the faster motion (1.6 sec cycle time) and the slower motion (2 sec cycle time).

2. When horizontal side loads were added to each finger, for both control methods the 1.6 sec cycle time led to an overshoot in F ; but this overshoot was less severe for Method 1 (16.7%) than for Method 2 (25.6%).

3. The time during which F exceeded 2% of its prescribed value was always less than 10% of its cycle time.

4. When the cycle time was increased to 2 sec (or longer, as indicated by other experiments), the effects of added side loads disappeared, and the accuracy required to maintain F was that of zero side load.

Not included in these eight experiments are those on repeatability, or the ability of the finger tip to return to an original horizontal position under the original set of internal finger pressures, while maintaining a constant gripping force. Results of these experiments show a maximum deviation of + 11% in the horizontal tip deflection δ between δ measured for a set of finger pressures as pressure increased, and δ measured for the same set of finger pressures as pressure decreased. Such deviations between loading and unloading cycles show the effects of material hysteresis which is always apparent to some degree in polymeric materials such as polyurethane.

From this two-part study, we conclude that the flexible arm manipulators and grippers are fast-acting, robust, and practical. Typical laboratory-scale versions have ratios of payload to self-weight in the range from one to five. On a larger scale, two-fingered grippers resemble a pair of controlled arm manipulators or tentacles and may be useful in construction projects, for instance, for gripping and manipulating objects too heavy for humans to handle.

Acknowledgements

Part 1 was sponsored in part by the U.S. Defense Advanced Research Projects Agency under Contract No. MDA903-84-C-0243 to

Duke University. Part 2 was sponsored in part by Lord Corporation under a consulting contract to the senior author. The authors appreciate the experimental assistance of Richard Lazarus and Glenn Butcher; and thank Jack Rebman, project monitor of Lord Corporation, for suggesting the two-fingered gripper design.

Table 1 Typical Experimental Results for Placement Accuracy of the Manipulator Arm

PROG.	T sec	T1 ms	T2 ms	ΔX mm	ΔY mm	$\Delta X/X$ $\times 100\%$	$\Delta Y/Y$ $\times 100\%$	$\Delta \theta$ deg	A cm
I	16.4	70	210	+2	+2.5	+0.43	+5	22	4
II	4.2	105	0	+4	+4.5	+0.85	+9	15	7

Table 2 Typical Experimental Results for a Two-Fingered Gripper in Harmonic Motion While Grasping a Block

Imposed Conditions (block displacement: ± 1.9 cm)				Measured Gripping Force F (N)	
Cycle Time (sec)	Side Load (N)			Control System 1	Control System 2
	Finger A	Finger B			
1.6	0	0		8.1 ± 0.14	8.1 ± 0.16
2	0	0		8.1 ± 0.14	8.1 ± 0.16
1.6	1.42	0.86		8.1 ± 1.35	8.1 ± 2.07
2	1.42	0.86		8.1 ± 0.14	8.1 ± 0.16

References

1. Backe, W. The application of servopneumatic drives for flexible mechanical handling techniques. Robotics and Autonomous Systems. Vol 2, pp. 45-56. 1986

2. Cutkowsky, M. R. Mechanical properties for the grasp of a robotic hand. Report No. CMU-RI-TR-84-24, Robotics Institute, Carnegie-Mellon Univ. Pittsburgh, Pa. 1984
3. Dario, P. and Buttazzo, G. An anthropomorphic robot finger for investigating artificial tactile perception. *Int. J. of Robotics Res.* Vol. 6, pp. 25-48. 1987
4. Datseris, P. and Palm, W. Principles on the development of mechanical hands which can manipulate objects by means of active control. ASME paper No. 84-DET-37. 1984
5. Ghattas, G. Mechanics of flexible, pneumatic structural elements for robotic limbs. MS Thesis, Duke U., Durham, NC. 1988
6. Hanafusa, H. and Asada, H. Stable prehension by robot hand with elastic fingers. *Proc. 7th Int. Symp. Indus. Robots.* Tokyo. 1977
7. Iberall, T. The nature of human prehension: three dextrous hands in one. *IEEE Int. Conference on Robotics and Automation* Vol.1, pp. 396-401. 1987
8. Jacobson, S. C., Wood, J. E., Knutti, D.F., and Biggers, K. B. The Utah/ MIT dextrous hand: work in progress. *Int J. of Robotics Res.* Vol. 3 pp. 21-50. 1984
9. Kerr, J. and Roth, B. Analysis of multifingered hands. *Int. J. of Robotics Res.* Vol. 4, pp. 3-17. 1986
10. Kier, W. M. and Smith, K. K. Tongues, tentacles and trunks: biomechanics of movement in muscular hydrostats. *Zool. J. Linnaea Soc.* Vol. 83 pp. 307-324. 1985
11. Kumar, V. and Waldron, K. J. Suboptimal algorithms for force distribution in multifingered grippers. *IEEE Int. Conf. on Robotics and Automation* Vol. 1, pp. 252-257. 1987
12. Lee, J. Kinematic synthesis of industrial robot hand/gripper -a creative design approach. *Robots and Autonomous Systems.* Vol. 4, pp. 257-263. 1988
13. Li, Z. and Sastry, S. Task oriented grasping by multi-fingered robot hands. *IEEE Int. Conf. on Robotics and Automation.* Vol. 1, pp. 389-394. 1987
14. Nguyen, Van-Duc. Constructing stable grasps. *Int. J. of Robotics Res.* Vol. 8, pp. 26-37. 1989
15. Palaniappan, M. Large deflections of continuous elastic structures. MS Thesis, Duke Univ., Durham, NC. 1986
16. Salisbury, J. K. and Craig, J.J. 1982. Articulated hands: force control and kinematic issues. *Int. J. of Robotics Res.* Vol. 1. 1982
17. Snyder, J. M. and Wilson, J. F. Dynamics of the elastica with end mass and follower loading. *J. Appl. Mech.* Vol 56, 1990
18. Wilson, J. F. Robotic mechanics and animal morphology. *Robotics and Artificial Intelligence*, M. Brady et al, eds., Springer-Verlag, NY. pp. 419-443. 1984a
19. Wilson, J. F. Active control of a two-finger manipulator. Lord Corp. Report, School of Engineering, Duke Univ. Durham, NC. 1984b
20. Wilson, J.F. Compliant robotic structures, Part III. U.S. Dept. of Defense, ADA192426, NTIS, Springfield, VA. 1987
21. Wilson, J.F. and Mahajan, U. The mechanics and positioning of highly flexible manipulator limbs. *J. Mechanisms, Transmissions, and Automation in Design.* Vol.111, pp. 232-237. 1989
22. Wilson, J.F. and Snyder, J. M. The elastica with endload flip over. *J. Appl. Mech.* Vol. 55, pp. 845-848. 1988

Progress in the Design and Control of Pseudomuscular Linear Actuators

G. Casalino ³, P. Chiarelli ^{1,2}, D. De Rossi ^{1,2},
G. Genuini ^{1,2}, P. Morasso ³, M. Solari ³.

¹ Centro "E. Piaggio", Università di Pisa, Italia

² Istituto di Fisiologia Clinica del CNR di Pisa, Italia

³ Dipartimento di Informatica, Sistemistica e Telematica
(DIST), Università di Genova, Italia.

Abstract

This paper provides a description of the functional basis of linear "soft" actuators based on polymer contractility elicited by electrochemical phenomena and driven by external electrical stimuli

Ion transport and motion equations describing the dynamic behavior of the actuator are presented. The problem of model complexity reduction is also considered with the aim of obtaining simpler, but still reliable models suitable to be used in the planning and control of articulated chain movements. Finally, a dynamic model for the control of a simple kinematic chain is presented based on a recruitment scheme.

1. Introduction

In the early fifties the capacity of directly converting chemical free energy into mechanical work by certain classes of synthetic polymers was discovered and analyzed.

Since then, mechanochemical polymeric systems have been investigated with the aim of constructing new motors, fundamentally different from the conventional electric and thermo-mechanical ones (1).

The polymeric actuators form a new class of devices characterized by their own weight to power ratio, attainable degree of miniaturization and built-in compliance.

This class of devices is thought to be potentially important in various fields of application, such as advanced robotics, prosthetics and artificial organ technology.

A polymer based pseudomuscular actuator generally consists of a bundle of fibers or thin strips of a weakly ionized polyelectrolyte gel, grouped together in a suitable way. The cross-linked polyelectrolyte is responsible for the dynamic properties of the actuator; contraction and relaxation can be induced by addition of suitable chemicals that permeate the gel and react reversibly with the polymeric matrix. Unfortunately, such a technique implies high energy dissipation and is impractical for most potential applications.

A further step, in order to implement more easily controllable devices, is represented by the possibility of electrically driving these actuators; this method has been actively explored and several electrically driven actuators have been proposed.

Basically they can be divided in two groups: the electrolytically (2,3) and electrokinetically activated actuators (4,5).

However, these systems have also shown limitations to their practical use, owing to the generation of electrolysis gases and slow kinetics for the former, and small force per unit of cross-sectional area for the latter.

More recently the preparation of blends of contractile and electron conducting polymers, which could possibly overcome some of the limitations above, has been reported (6).

The peculiar functional and structural properties of the polymeric actuators and their complex dynamic behavior also demand for new control strategies. A few attempts in this area have been reported (7,8), but the need still exists for more accurate and efficient ways of activating and controlling electromechanochemical polymer actuators.

In this paper we report our progress in the implementation of polymer actuators and their modelling and control in the intent of developing an anthropomorphic finger-like articulated chain.

2. Structure and Characteristics of the Actuator

The structure of the actuator we have conceived is depicted in Fig.1.

The stimulating carbon fiber electrodes (Courtaulds Grafil Ltd, U.K., 7 μm in diameter) and the contracting gel fibers form a parallel bundle located inside a rubber container filled with a diluted NaCl aqueous solution; a platinum counter electrode closes the electrical circuit.

The contractile gel used here is polyacrylonitrile (PAN) in form of fibers (Courtaulds Grafil, Ltd, U.K., 13 μm in diameter) cross-linked and ionized to obtain a polyelectrolyte gel structure, following the procedure suggested by Itoh et al. (9).

This configuration of the actuator, with tightly packed contractile and stimulating fibers in parallel, provides both strong forces and fast response. Minimizing the electrodiffusion path of the reactants, generated by electrolysis at the carbon fiber-solution interface, to the contractile fibers and inside them, renders much shorter the actuator response time which is, as a first approximation, proportional to the square of the interfiber distance and contractile fiber diameter (10).

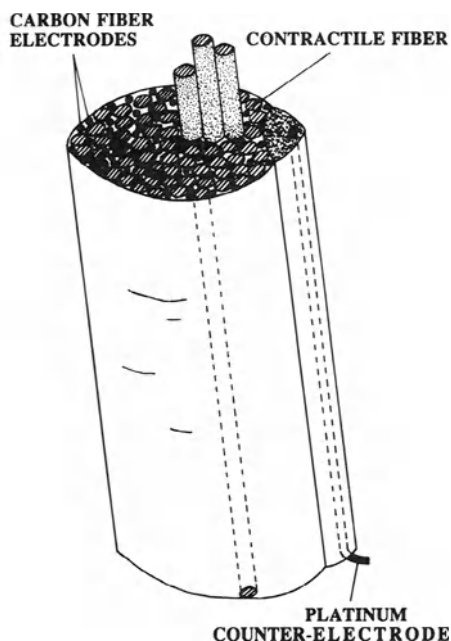


Figure 1. A schematic drawing of the actuator : contractile polyacrylonitrile gel fibers and carbon fibers are arranged in parallel .The carbon fiber electrodes are grouped in bundles and connected by a multiplexer to the power supply. A part, a platinum counter electrode is in contact with the electrolytic bath to close the electrical circuit.

In addition, the flexibility and low weight of the carbon fibers provide minimum mechanical impediment to contraction and a favourable weight/power ratio to the actuator.

As already proven (3,6), electrically activated contraction results to be slower than the chemically activated one (see Figs.2 and 3). The rate limiting factor under electrochemical stimulation of the polyacrylonitrile fibers is represented by the time to allow currents to flow throughout the system and free the equivalents of the chemical species involved in the process;

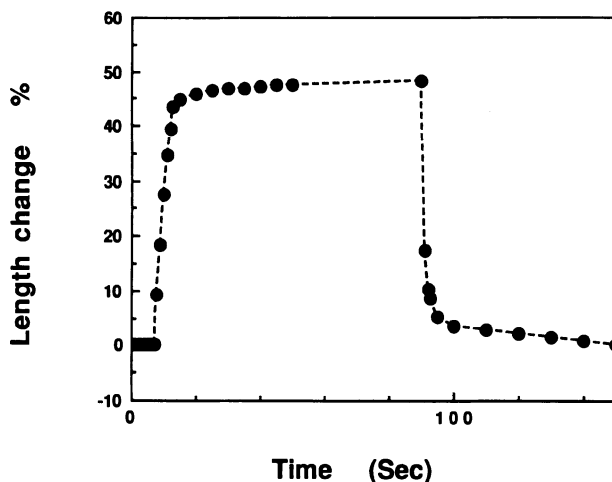


Figure 2. The isotonic fractional length change of the polyacrylonitrile gel fibers as a function of time when the pH is suddenly changed from 1 to 13 and viceversa.

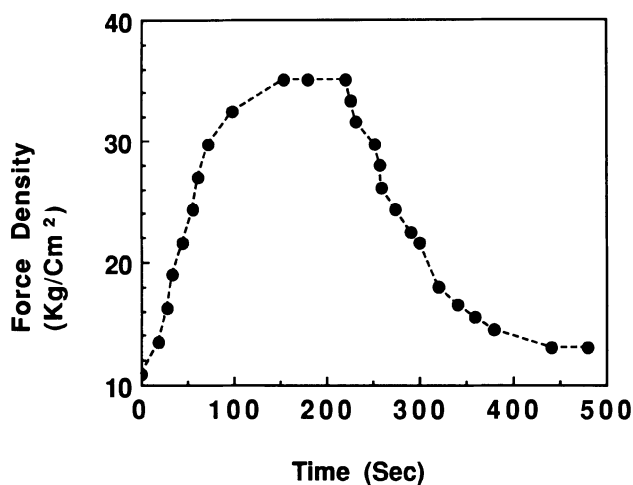


Figure 3. The isometric force density exerted by polyacrylonitrile gel fibers as a function of time when stimulated by carbon fiber electrodes with a potential step of +10 volts and -10 volts respectively (referred to the platinum counterelectrode).

since this fact can require a large current in a short time, it couples the contraction response time of the system to its electrical resistance. Moreover, gradients of pH, generated by a electrical stimulation, are limited in magnitude.

The actuator architecture, illustrated in Fig.1, is thought in order to obtain an electrically driven actuator with speed of contraction and relaxation and related forces comparable with those obtained by changing the bath composition by rapid fluid pumping.

The isotonic and isometric response of a PAN bundle to respectively chemical and electrochemical stimuli are reported in Figs. 2 and 3.

The observed maximum isotonic length change is of the order of 50%, when the pH is changed from 1 to 13 and viceversa.

The measured contraction and relaxation times of these fibers are of the order of 1 second.

Under electrochemical stimulation the isometric force generated by a fiber bundle has been found to be about 20 Kgp/cm²; such a force density is comparable or even higher than the one developed by the fibers of mammalian muscles.

3. Coupled Phenomena and Rate Limiting Factors

The contraction mechanism of a polyelectrolyte gel is well understood in terms of mechanochemistry : a change of a chemical variable such as pH, or salt concentration, forces the gel to undergo a structural rearrangement which changes its volume (11,12). In the case of electrochemical activation, the main aspects involved are the following:

- the electrochemical reaction at the electrodes, that causes the growth of acid or basic fronts at their surfaces (depending on electrical polarity);
- the propagation of these fronts and other chemical species in the solution;
- their diffusion inside the gel fibers;
- the mechanochemical readjustment of the polymeric network

If the diffusion time of the chemical gradients inside the gel fibers is comparable with the polymeric network readjustment time, these two phenomena are strictly coupled.

This fact implies a great difficulty both in the study of the contraction kinetics and in the planning of a control strategy. On the contrary, when the mechanical response time of the fiber is very different from the diffusion time of the ions, the two phenomena can be decoupled.

The possibility of decoupling the motion of the stimulating ions from the structural rearrangement of the cross-linked polymeric chain can be verified by analyzing the two phenomena separately: an analysis of the diffusion-reaction phenomena which take place into an ionizable gel, when the pH of the bath changes, has been given by Nussbaum and Grodzinsky (13) and an expression for the limiting diffusion-reaction time constant τ_{mc} has been derived when the pH of the external baths is steeply increased of small amount.

The time constant of the mechanical readjustment of the polymer network in form of fiber is calculated to be $\tau_m = d^2 f / (4\alpha_0^2 \pi^2 \mu)$ (14), where f is the polymer-solution friction coefficient, μ is the shear elastic modulus of the gel, d is the fiber diameter and α_0 is the first root such that $J_0(\alpha_0 \pi) = 0$, where J_0 is the zero order regular Bessel function ($\alpha_0 \approx 0.767$).

For strong gels, with elastic moduli of the order of 10^7 C.G.S. in the wet state and with a friction coefficient of 10^{13} C.G.S. and a pKa of 5 it is possible to decouple the problem in a large interval of pH (pH > 2) (15).

A dual analysis has been done with respect to the kinetics of the gel network readjustment (10). These condition holds also for PAN and, hence, the polymer at any time can be considered to be in mechanical equilibrium at the local proton concentration, the kinetics being limited by diffusion - reaction phenomena. It follows that actuator response can be analyzed by a mechanical dynamic model whose material parameters, characterizing the force-displacement

relation, are modulated in space and time by the electrochemical stimulus. In the following paragraph a concentrated parameter model of the gel dynamic response is formulated.

4. Lumped Parameter Model of the Mechanical Behavior of a Single Gel Fiber

A continuum model was first developed and utilized both to describe the free swelling kinetics of the gel and to predict its force response to a mechanical input. (7)

The continuum approach is useful to determine the local stress in the gel. Since we are primarily interested in controlling the dynamic response of the actuator, some complexity reduction has to be performed.

A lumped parameter model, suitable to represent the viscoelastic behavior of a gel fiber, is the standard linear viscoelastic solid, depicted in Fig. 1, which results to be valid at least when the intrinsic viscoelastic behavior of the solid component can be neglected.

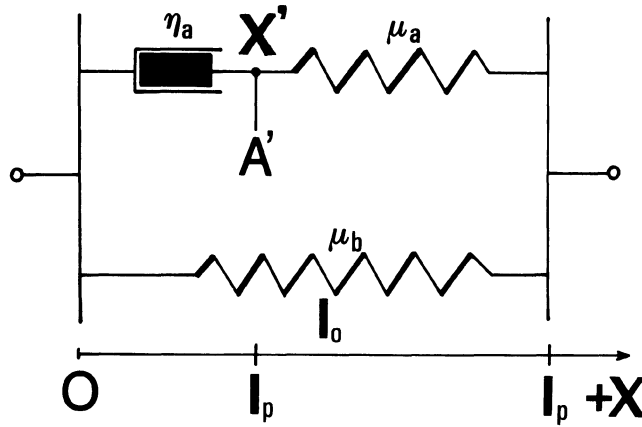


Figure 4. Schematic drawing of the standard linear solid model reduction of the gel fibers.

Referring to Fig.4, the forces generated by the standard linear viscoelastic solid can be expressed as

$$F_a = (\mu_a S/l_0) (l_p + x - x') \quad [1.a]$$

$$-(\eta_a S/l_0) (x') + F_a = 0 \quad [1.b]$$

$$F_b = (\mu_b S/l_0) (l_p + x - l_0) \quad [1.c]$$

$$F = F_a + F_b \quad [1.d]$$

where l_0 is the unstrained length, l_p the pretension length, x is the instantaneous length of the gel fiber, and x' is the fictitious instantaneous length at the intermediate point A' . Moreover S

is the resisting cross section of the contractile fiber per-unity length/section and μ_a, μ_b, η_a , respectively are the elastic and viscous moduli .

The dependence of the parameters, just defined with the help of Fig.4, from the physical gel constant of the continuum model, can be deduced knowing that: following an instantaneous step deformation, the behavior of the gel at time instant $t = 0^+$ is similar to an incompressible medium since the deformation can be assumed to take place at constant volume (16).

Thus at time instant $t = 0^+$, equating the force of the linear viscoelastic solid, submitted to a step elongation, with that of the gel (whose polymeric component has a bulk modulus k , a shear modulus μ and the polymer-solution friction coefficient is f) we have (16):

$$\mu_a + \mu_b = 3 \mu \quad [2]$$

Also equating, at the time instant $t = \infty$, the asymptotic force value of the linear viscoelastic solid with that of the gel we have

$$\mu_b = [9 (k / \mu) / (1 + 3(k / \mu))] \mu \quad [3]$$

Then, substituting equation [3] into [2] it follows

$$\mu_a = [3 / (1 + 3(k / \mu))] \mu \quad [3]$$

Finally, by equating the stress-relaxation decay time constant τ_m of the gel with that of the linear viscoelastic solid, that is

$$\tau_m = (d^2 / 4\alpha_o^2 \pi^2) (f/\mu) = (S / \alpha_o^2 \pi^3) (f/\mu) = \eta_a / \mu_a \quad [4]$$

an expression for the viscous coefficient η_a can be obtained as

$$\eta_a = (S / \alpha_o^2 \pi^3) [3 (f/\mu) / (1 + 3(k / \mu))] \mu \quad [4']$$

The expression [4'] can be reasonably assumed to hold for the general case of arbitrary time history force.

It is worth noting the fact that the ratios k/μ and f/μ appearing in [3], [3'], [4], [4'] can be regarded as substantially independent from the pH of the gel, since both k and f result to be practically proportional to μ as discussed in (10).

Then, as a consequence of the above practical assumption, it is not difficult to realize that the problem of representing the dynamic behavior [1a -1d] of the fiber, for time varying pH, substantially relies upon a characterization of the sole terms $(\mu S)/l_o$, μS and S as functions of the gel pH itself.

Although the reduction of a contractile gel fiber to a viscoelastic solid with tunable parameters may loose some of the complexities of the real system, it makes however possible to study control techniques for complex multi-fiber actuators working in agonist-antagonist configurations on kinematic joints.

In the next paragraph the features of a recruitment method, when applied to the control of a single joint, is discussed in some detail.

5. Recruitment Control for the Motion of Articulated Chains

In a previous work (7) we have investigated the possibility of using the pseudomuscular actuators to move articulated chains.

In that occasion we have focused our attention on a push-pull control configuration; in this section, for reasons we are going to explain, we look for the possibility of adopting a recruitment control configuration.

The present analysis is performed with reference to the structure depicted in Fig. 5 where the joint is characterized by the presence of two multifiber actuators each one representing the agonist and antagonist muscle respectively.

On the basis of relationships [1a - 1d] (valid for a single fiber) and also keeping into account expressions [3], [3'], [4], [4'] it is not difficult to verify (see however also (7)) that the net torque generated at the joint by a couple of a single agonist-antagonist fibers can be expressed (approximately) as:

$$m = -p[(\mu S/l_0)^{(1)} + (\mu S/l_0)^{(2)}] \theta - q[(\mu S/l_0)^{(1)} + (\mu S/l_0)^{(2)}] \alpha + u \quad [5.a]$$

$$u = s\{[l_p(\mu S/l_0) - \mu S]^{(1)} - [l_p(\mu S/l_0) - \mu S]^{(2)}\} \quad [5.b]$$

where the upscripts "(1)" and "(2)" refer to the agonist and antagonist fiber respectively and the (practically constant) terms p , q , s are given by

$$p = [9 r^2 C_{k\mu} / (3C_{k\mu} + 1)] \quad ; \quad C_{k\mu} = k / \mu \quad [5.c]$$

$$q = (d^2/\alpha_0^2 \pi^2) [3 r C_{f\mu} / (3C_{k\mu} + 1)] \quad ; \quad C_{f\mu} = f / \mu \quad [5.d]$$

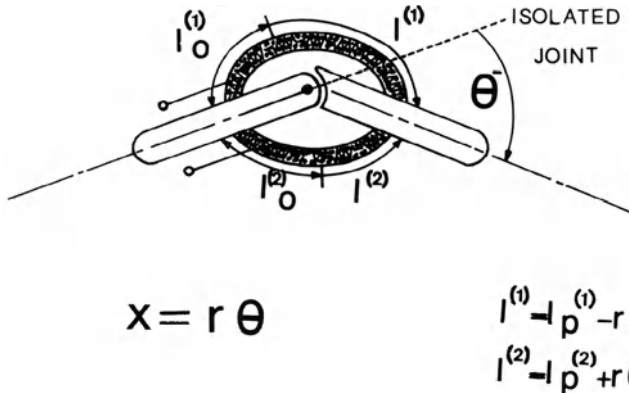


Figure 5. Schematic drawing of the agonist-antagonist couple of multifiber actuators at the joint.

$$s = p / r \quad [5.e]$$

with r representing the radius of the joint. Moreover θ is the angle of rotation of the joint and α ("filtered velocity") the output of the differential equation

$$\tau_m \alpha + \alpha = r \theta \quad [6.a]$$

having time constant τ_m equal to the "average" value of τ_m (the time constant of the gel, see [4]) and representing an (approximate) aggregate model for the viscous effects existing in the couple of fibers.

Relationships [5.a] and [5.b] bring into evidence the fact that action m , produced at the joint, by a generic single couple of agonist-antagonist fibers, can be considered equivalent to that given by an external signal u acting together with a (tunable) position and filtered velocity feedback (being the velocity filtered by equation [6.a]).

Let us now pose, for ease of notation

$$w = \mu S / l_0 \quad [7.a]$$

$$z = l_p \mu S / l_0 - \mu S \quad [7.b]$$

These new variables actually exhibit a qualitative sigmoidal behavior (as the one depicted in Fig.6) as a consequence of the fact that the same occurs for both terms and μS . Then the torque generated by a single couple of fibers can be rewritten as

$$m = -p[w^{(1)} + w^{(2)}] \theta - q[w^{(1)} + w^{(2)}] \alpha + u \quad [8.a]$$

$$u = s[z^{(1)} - z^{(2)}] \quad [8.b]$$

Assuming that the agonist and antagonist actuators are composed by a number N of fibers, of which, $\lambda_i N$ ($\lambda_i < 1$), excited at the maximum level (w_{max} , z_{max}) and $(1 - \lambda_i) N$ excited at the minimum level (w_{min} , z_{min}), we have that the torque generated at the joint results to be

$$M = -N p [2 w_{min} + (\lambda_1 + \lambda_2) \Delta w] \theta - N q [2 w_{min} + (\lambda_1 + \lambda_2) \Delta w] \alpha + U \quad [9.a]$$

$$U = N s \Delta z (\lambda_1 - \lambda_2) \quad [9.b]$$

where

$$\Delta w = w_{max} - w_{min} \quad [10.a]$$

$$\Delta z = z_{max} - z_{min} \quad [10.b]$$

Then, by imposing

$$\lambda_1 = \lambda_0 - \delta \lambda \quad [11.a]$$

$$\lambda_2 = \lambda_0 + \delta \lambda \quad [11.b]$$

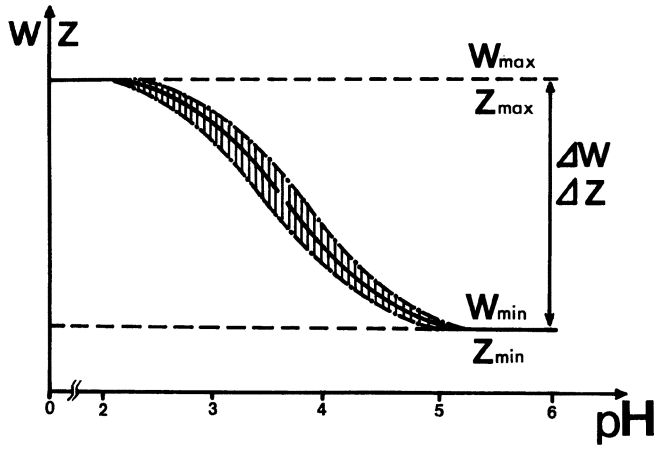


Figure 6. A qualitative representation of the function w, z , as a function of the pH inside the contractile gel fibers .

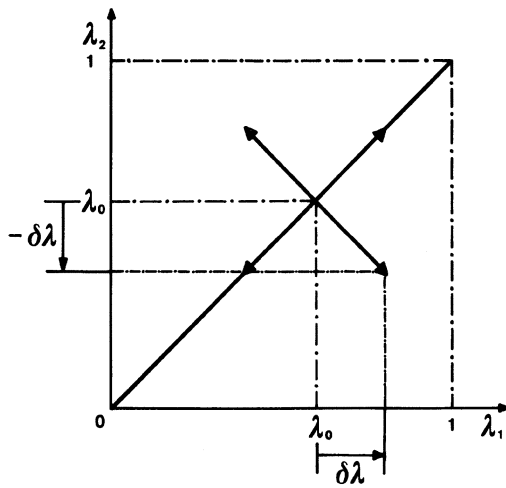


Figure 7. Recruitment control scheme: on the X and Y axes there are the numbers λ_1 and λ_2 of excited fibers in the agonist and antagonist muscle; $\lambda_0 = (\lambda_1 + \lambda_2)/2$ and $\delta\lambda = (\lambda_2 - \lambda_1)/2$ govern the stiffness and the torque of the joint respectively.

we get

$$M = -K_p(\lambda_0) \theta - K_v(\lambda_0) \dot{\alpha} + U \quad [12.a]$$

$$U = G \delta \lambda \quad [12.b]$$

where

$$K_p(\lambda_0) = 2 N p [w_{min} + \lambda_0 \Delta w] \quad [13.a]$$

$$K_v(\lambda_0) = 2 N q [w_{min} + \lambda_0 \Delta w] \quad [13.b]$$

$$G = 2 N s \Delta z \quad [13.c]$$

which tell us that the elastic and viscous coefficients K_p and K_v of the muscles linearly depend on the bias λ_0 assigned to the number of fibers excited at the maximum level when the total arrangement is in its rest position.

By modulating in an opposite manner the number of electrically activated fibers in both muscles, it is possible to assign the reference signal U to the global actuation system while maintaining constant the values of the feedback parameters K_p , K_v .

A great advantage of this strategy control technique, if compared with the one considered in (7), is also represented by the stability of K_p and K_v which depend only on the saturation values of the z and w curves, and are therefore not affected by uncertainties existing in the middle range of the above mentioned curves represented in Fig.7.

Finally, the possibility of quite easily fixing the value of K_p and K_v (by simply changing λ_0), also opens some perspectives in the field of the so called "ibrid control", where not only the positions, but also the interaction forces with the manipulated objects are requested to be controlled.

5. Conclusions

Many efforts are still required to implement a stable and long life device, which responds quickly to external stimuli and which has gentle and precise movements.

To satisfy all these needs, thinner contractile fibers made of strong, fatigue resistant gels, new electrode materials, and assembling technique, must be available.

Although they are not available yet, the peculiar characteristics of a polymeric actuator make it a promising tool for applications in many fields such as micromanipulation, robotics, prosthetic, and artificial organs.

For this reason the control of a polymeric linear actuator has an interesting theoretical and practical appeal.

The next step in the work will be the characterization of the polymeric (PAN) component of the actuator through the measure of its material parameters defined by the analysis above.

Acknowledgment

This work is supported in part by the "Progetto Finalizzato Robotica" of the National Council of Research of Italy under the contracts n° 89.00507.67 and n° 89.00499.67.

References

1. Osada Y.: Conversion of chemical into mechanical energy by synthetic polymers (chemomechanical systems), *Adv. Polym. Sci.*, 82, 1-46 (1987).
2. Hamlen R.P., Kent C.E. and Shafer S.N.: Electrolytically activated contractile polymers, *Nature*, 206, 1149-1150 (1965).
3. De Rossi D., Chiarelli P., Buzzigoli G., Domenici C., Lazzeri L.: Contractile behavior of electrically activated mechanochemical polymer actuators, *Trans. Am. Soc. Artif. Intern. Organs*, XXXIII, 157- 162 (1986).
4. Osada Y., Hasebe M.: Electrically activated mechanochemical devices using polyelectrolyte gels, *Chemistry letters*, pp.1285-1288 (1985).
5. Grodzinsky A.J., Melcher J.R. : Electrochemical transduction with charged polyelectrolyte membranes, *IEEE Trans.Biomed. Eng.* BME-23, 421-433 (1976) .
6. Chiarelli P. , Umezawa K., De Rossi D.: A polymer composite showing electrocontractile response , to be published on : *POLYMER GELS: Fundamental and Biomedical Applications*, Plenum Publishing Company L.t.d., London.
7. Genuini G., Casalino G., Chiarelli P., De Rossi D.: Pseudomuscular linear actuators: Modelling and simulation experiences in the motion of articulated chains, *NATO Workshop* , October 25-31 1987, Il Ciocco, Italy. To be published as a book in the NATO ASI Series.
8. Caldwell D.G., Taylor P.M.: Chemically Stimulated Pseudo-Muscular Actuation, *Int. Journal of Engineering Science*. In press .
9. Itoh Y., Matsumura T., Umemoto S., Okui N., Sakai T. : Contraction/elongation mechanism of acrylonitrile gel fibers, *Polymer Preprints-Japan (English Edition)*, 36(Nos.5-10), p. E184 (1987).
10. Chiarelli P., De Rossi D.: Determination of mechanical parameters related to the kinetics of swelling in an electrically activated contractile gel, *Progr. Colloid Polym. Sci.* 78; 4-8 (1988).
11. Tanaka T.: Gels, *Sci. Am.* 244, 110-124 (1981).
12. Kuhn W., Ramel A., Walters D.: Conversion of chemical into mechanical energy by homogeneous and cross-striated polymeric system, from *Size and Shape Changes of Contractile Polymers* edited by A. Wassermann, Pergamon Press (1960), pp.41-77
13. Nussbaum J.H., Grodzinsky A.G.: Proton diffusion reaction in a protein polyelectrolyte membrane and the kinetics of electromechanical forces, . *J. Membrane Science* 8, 193-219 (1981).
14. Chiarelli P., unpublished.
15. Chiarelli P., Genuini G., De Rossi D.: Electrochemically driven muscle-like actuators. *Proc. of IX Conf. IEEE Eng. in Medicine and Biology*, vol 2, 1089-1091, (1987).
16. Chiarelli P., Basser P., De Rossi D., Goldstein S.: Dynamics of hydrogel strip, *Biorheology*. In Press.

Shape Memory Alloy Linear Actuators for Tendon-based Biomorphic Actuating Systems

M.Bergamasco, F.Salsedo, P.Dario

Scuola Superiore S.Anna, Pisa

and

Centro "E.Piaggio", University of Pisa, Pisa, Italy

Abstract

This paper describes the different design aspects to be considered for the development of Shape Memory Alloys (SMA) actuators. The design of the actuator has been addressed first by a deep analysis of the thermomechanical properties of SMA material. The different components of a SMA actuator have been presented and their role in the economy of the whole actuation system has been outlined. The description of the experimental prototype realized in our laboratory is given together with the presentation of the results of the first step of experimental tests.

The favourable feature of obtaining ad hoc shaped active elements from SMA material is emphasised in order to design elementary active structures to be utilized for the realization of robotic actuators for future anthropomorphic robot hands.

1 Introduction

Present research on actuation systems for robot effectors is markedly characterized by the effort of developing new technologies for actuators and motors. This trend takes its origin from the analysis of actual performances

of actuators, based on conventional technologies, that are currently utilized in robot effectors design. Conventional actuators introduce practical design constraints especially in terms of weight and encumbrance; the development of manipulators or artificial arms possessing effective slenderness and dexterity is strongly restrained by the unfavourable values of power-volume/weight ratios of present actuator technologies. In this paper we present the analysis performed in our laboratory of a new actuator technology, i.e. that of Shape Memory Alloys (SMA). The motivation at the origin of the choice of SMA technology for robot actuators refers to the fact that SMA represent, at present, a borderline between the aspects of basic research experimentation and immediate technical applicability. We believe that further research efforts on Shape Memory Alloys are needed in order to achieve an effective reliable technology for actuators in general, but the experimental results we obtained, and that are described in the following, at a first stage of experimentation have demonstrated their immediate applicability under defined working conditions.

1.1 Shape Memory Alloys

SMA are alloys capable of recovering, when heated over a determined transition temperature, the large deformation impressed at low temperature. The Shape Memory Effect (SME) was discovered in 1938 by A.B. Greninger (Harvard University) and V. G. Moradian (MIT), who demonstrated temperature induced shape changes in brass. In 1962 M. J. Bruehler of U.S. Naval Ordnance Laboratory obtained a Nickel-Titanium alloy (called Nitinol) whose performances in shape recovering were higher with respect to other materials. The Shape Memory Effect depends on the transformation occurring between two solid phases possessing different mechanical properties. The austenite phase, which is stable at high temperature, is characterized by an elastic behaviour until the "plastic stress" of the material. The martensite phase is stable at low temperature and possesses a plastic behaviour: by applying low forces it is possible to obtain large permanent deformations. When strains in the martensite phase do not overcome a limit ϵ_L , characteristic for each SMA alloy, it is possible to completely recover the original unstrained shape by heating the material over a determined transition temperature. The heating operation activates the reverse martensite-austenite transformation. The percentage of martensite versus temperature is reported in Fig.1 for a SMA

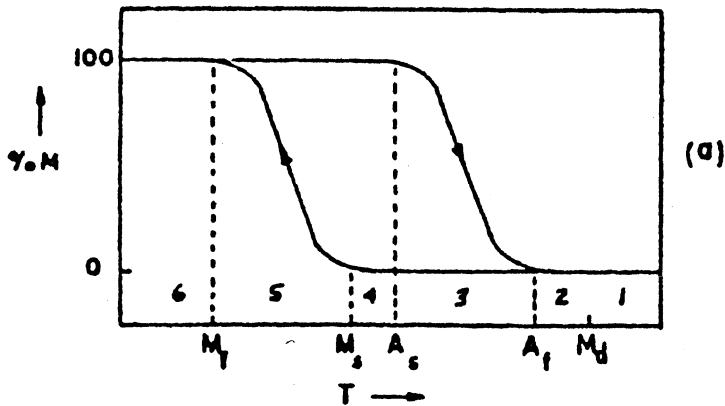


Figure 1: Transition temperatures for a SMA sample (from J. Perkins et al.)

sample; the different transition temperatures are also shown.

There are four transition temperatures: two of them, A_s (Austenite start) and A_f (Austenite finish), regulate the martensite-austenite transformation, while M_s (Martensite start) and M_f (Martensite finish) refers to the inverse austenite-martensite transformation. The general phase transformation is characterized by a large hysteresis. The four temperatures A_s , A_f , M_s and M_f can be largely varied (from -100°C to 200°C for Nitinol) by appropriately tuning the percentage of alloy components, the thermal treatment of alloy components, the thermal treatment or also the moulding process of the semi-finished material. The initial shape of the SMA sample can be obtained by means of an annealing process: the ultimate mechanical characteristics of the material depend on such annealing parameters as annealing temperature, annealing time and cooling rate.

Shape Memory Effect can operate only if the alloy can exhibit a martensitic transformation, i.e. a reversible transformation in which the stable phase at low temperature (martensite) is generated from the stable phase at high temperature (austenite or parent phase) simply by lattice re-ordering (without diffusion of atoms). A typical example is that of the ternary alloy Cu-Zn-Al where the austenite possesses a cubic centered structure, while martensite is orthorhombic. By the crystallographic analysis of the martensite, it is possible to locate some variants where the crystal is oriented according to a determined characteristic direction. For the Cu-Zn-Al alloy, four variants are present. In the case there are no external loads on the sample (zero stress), the variants are homogeneously distributed and at the macroscopic

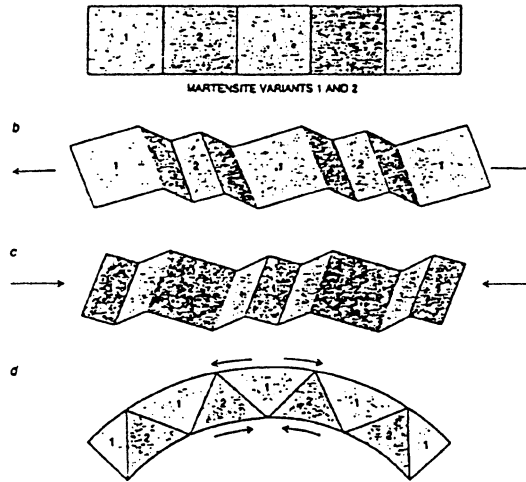


Figure 2: Growing of martensite variants in a SMA sample

level it is not possible to distinguish a preferential orientation of martensite. When the external load is applied, the martensite interacts with it by growing some variants with respect to others.

This phenomenon can be depicted, for the case of only two martensite variants, as in Fig.2: in a) the SMA material is characterized by an homogeneous distribution of martensite 1 and 2 variants; when the external load is applied (a tensile force, as in b)) variant 1 grows with respect to variant 2; the reverse situation occurs when the external load is a compression force (as depicted in c)); when , at the other hand, the material undergoes a flexion action (as in d)) the external fibres will be mainly constituted by variant 1, while internal fibers by variant 2 of the martensite. The growing of one variant involves the re- allocation at the border between this variant and the others in such a way to conform themselves to the orientation correspondent to the growing variant. The more common atoms re-allocation mechanisms for SMA alloys is that of gemination. The growing of the variants that are favourably oriented with respect to the applied load translates into a macroscopic deformation of the material. This deformation retains also when the external load is removed (pseudoplasticity of martensite). In order to completely recover the original shape of the sample, it is essential that the transformation being cristallographically reversible. This fact means that, during the inverse martensite-austenite transformation, the crystal could recover the orientation of the parent phase. Crystal reversibility is a general characteristic of thermoelastic transformations or of martensitic transformations of ordered alloys (i.e. alloys where atoms are arranged in preferential

places in the crystal structure). Another condition to be satisfied in order to obtain a complete shape recovery is that deformation mechanisms of martensite must be activated by forces lower than the real "plastic stress" of the material (which is strictly tied with irreversible motion of dislocations). This condition is satisfied for SMA alloys, where germination is activated by relatively small values of forces. Hence, the large "compliance" of the martensite can be correlated to the low force necessary to deform the SMA material by germination mechanism. Clearly this property of strongly interacting with the applied load affects the thermodynamic stability of the two phases. From a thermodynamics point of view this effect can be evaluated by considering a general expression by describing the free energy of the system (Otsuka, Shimizu, 1986) that could take into account the contribution of the work performed on the system by the external load. It has been demonstrated that in cases of "monoaxial stress" the applied load (both tensile or compression) promotes martensite stability. Therefore, a SMA alloy can be considered as a thermodynamic system, characterized by three status variables (temperature, stress and strain), possessing phase transition and very similar to the well known water-steam system. As a consequence the course of transition temperature versus applied load can be described by means of Clausis-Clapeyron equation. In Fig.3 the course of transition temperatures A_s and M_s versus the applied load is depicted for a Cu-Zn-Al alloy. The martensite transformation of SMA alloys can be activated by acting both on temperature (athermal martensite) and on stress (stress-induced martensite) (Perkins et al.,1975).

1.2 SMA phenomenology

The specific mechanical behaviour of SMA material is summarized in Fig.4.

When the temperature is lower than M_f , and the material is completely in the martensitic phase, the behaviour of the material is characterized by the high martensite pseudoplasticity. By increasing the strain, the stress initially elastically arguments according to the Young modulus of the martensite; however the elastic part of the curve is limited since the gemination strain of the martensite is activated by low stresses. At the end of the elastic course, the curve maintains an approximately constant value until the gemination strain is over. At this point the stress takes again an elastic course according to the Young modulus of the martensite. When the external load is removed,

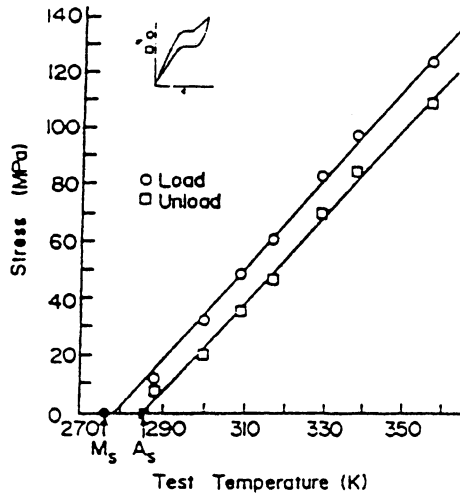


Figure 3: Transition temperatures A_s and M_s versus applied load (from Otsuka, Shimizu, 1986)

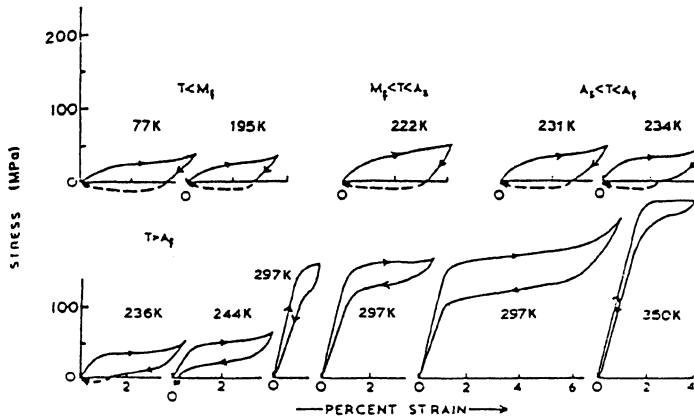


Figure 4: Stress-strain curves for Ch Zn Sn alloys (from Otsuka, Shimizu, 1986)

only the elastic portion of the strain can be recovered. In order to recover the plastic portion it is necessary to activate the inverse martensite-austenite phase transformation by heating the material at temperatures higher than A_f . In this field of temperatures the percentage of temperature recovered strain is considerable high (Shape Memory Effect, SME). The mechanical behaviour of the SMA material does not considerably vary until the temperature exceeds A_f . In this field of temperature, when stress is zero, the only austenite phase is present. When the strain increases, the stress increases according to the Young modulus of austenite until the activation threshold of the martensitic transformations is achieved; the correspondent stress proportionally increases with temperature (see Fig.3 already presented). At this stage the stress slightly increases by augmenting the strain, while, at the same time, the percentage of martensite (stress-induced martensite) increases. Stress-induced martensite (S.I.M.) is already oriented according to the applied load: for this reason, at the end of the martensitic transformation, the stress course follows the elastic behaviour according to the Young modulus of the martensite. By removing the load, at a first stage, the plastic strain of the martensite is recovered, then the portion of strain related to the inverse transformation martensite-austenite and, in the end, the elastic strain of the austenite. In this temperature range ($T > A_f$) pseudoplastic behaviour of the material is excluded, since to load removal corresponds a zero deformation of the sample (pseudoelastic behaviour or pseudo-elasticity, P.E.). For temperatures belonging to the range M_f - A_f the total strain recovery occurs for either Shape Memory Effect (S.M.E.) and also pseudo-elastic effect (P.E.) as depicted in Fig.5.

When the material undergoes a real plasticity an incomplete strain recovery occurs: we indicate this situation as a "loss of memory". In fact, when the critical sliding stress of the crystalline planes is achieved it is no more possible to completely recover the impressed strain even by heating the material at temperatures higher than A_f .

1.3 Applications of SMA in Robotics

Shape Memory Alloys can be utilized to generate displacements and/or exert forces against the external environment. Displacements can be generated by directly exploiting the SME, while large forces can be exerted by restricting the shape recovery by means of external constraints. In general it is possi-

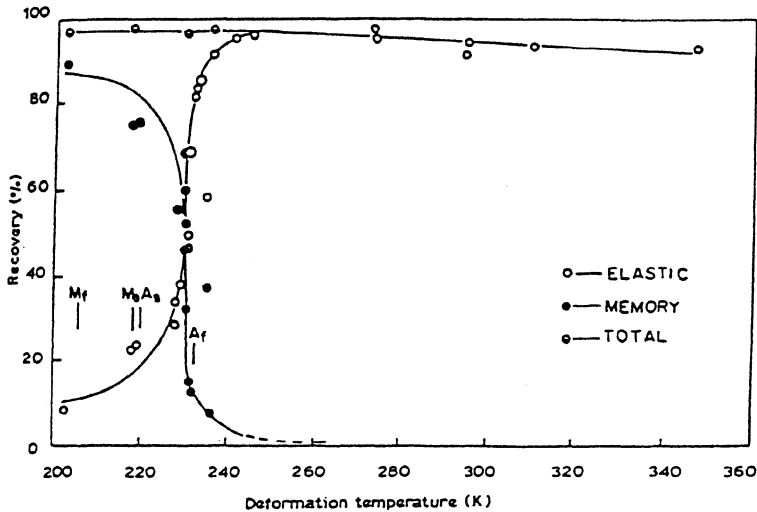


Figure 5: Percentage of recovered strain by Shape Memory Effect (S.M.E.) and Pseudo-Elasticity (P.E.)

ble to produce mechanical work by constraining the shape recovery of the material without completely blocking motion. With respect to mechanical performances, SMA show reduced volumes and weights. Moreover the possibility to correlate the control variable "temperature" with different types of electric variables represents a very effective feature in terms of real implementation of a direct control procedure: SMA are in fact characterized by fairly good values of resistivity hence allowing the heating of the material by means of Joule effect. On the other hand, a control variable such as temperature strongly affects the ultimate performances of SMA actuators. The rate of obtainable movements depends in fact from time constant of thermal transient. For this reason the utilization of adequate cooling systems is often required in order to achieve the desired bandwidth of the actuation system. Since SMA actuators can be seen as thermal machines, producing mechanical work from heating, their efficiency is extremely low: usually it does not exceed 10%. Then, from a general point of view with respect to existing conventional motors, SMA actuators must be considered as a complementary technology rather than a substitute one. In robotics, SMA have been utilized for both direct-drive type actuators (Fig.6) and linear actuators (Fig.7).

For both direct-drive and linear actuators, in order to obtain a reversible motion of the actuated joint, it is necessary to arrange an antagonistic element operating against the SMA active element. The antagonistic element can be passive (Fig.6a), e.g. a passive mechanical steel spring, or

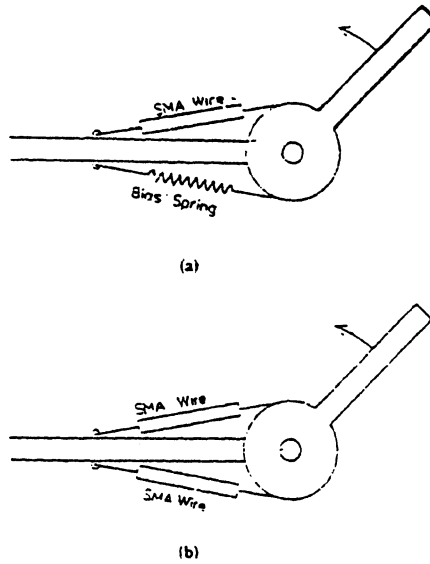


Figure 6: Direct-drive SMA actuators
(from S. Hirose et al. 1984)

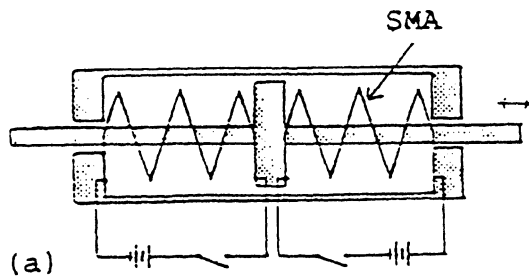


Figure 7: Linear SMA actuator
(from S. Hirose et al. 1984)

active (Fig.6b). Usually the antagonistic active element is another SMA element arranged in opposition with the other. This solution allows a better symmetry of the actuator system in terms of output mechanical performances. The SMA active elements can be wires (Hashimoto et al, 1985) (Nakano et al.,1984)(Gabriel et al.,1988) or helicoidal springs (Ikuta,Hirose et al,1988)(Yaeger,1984)(Hirose et al,1984). The electronics of the position and force servosystems is usually based on P.I.D. controllers for error signal processing and on P.W.M. drivers for the heating of the active elements (Hirose et al, 1987). Cooling systems of the active elements are based on air flow (Kuribayashi, 1986) or heat sinks (Hashimoto, 1985). The applications of SMA actuators are mainly focused on actuating systems for robot hands (Kuribayashi, 1986) or legged robots (M. Hashimoto, 1985).

2 SMA Active Elements

The active elements are SMA structures capable of generating, through shape recovery, the mechanical work utilized as the output of the actuator. They play a fundamental role in the design of the actuator system: in the course of our study we have considered the active elements as "structural elements" of SMA material possessing a pseudoplastic behaviour and also the capability of recovery a given deformation (strain). A SMA active element can be ideally considered as formed by two elementary components: a) the SMA material per se and b) its structural geometry. The "material" component is critical in order to obtain the desired mechanical performances of the active element: material characteristics such as recoverable strain, yielding stress, transition temperatures, resistivity, etc., must be carefully chosen in order to obtain the desired output. In our research work, however, it has not been possible, cause the reduced availability of SMA material, to test samples with different characteristics. The available "material" component of the active element has been carefully characterized in terms of its thermo-mechanical behaviour. To this aim different test typologies have been identified for wire samples undergoing tensile load. Tests on SMA material have been performed by having in mind the two general working conditions occurring for the actuators of robot effectors; since our primary aim was the development of actuator systems for robot hands, we referred in a particular to "stable grasp" and "manipulative" operations. During grasping procedures the actuators of a

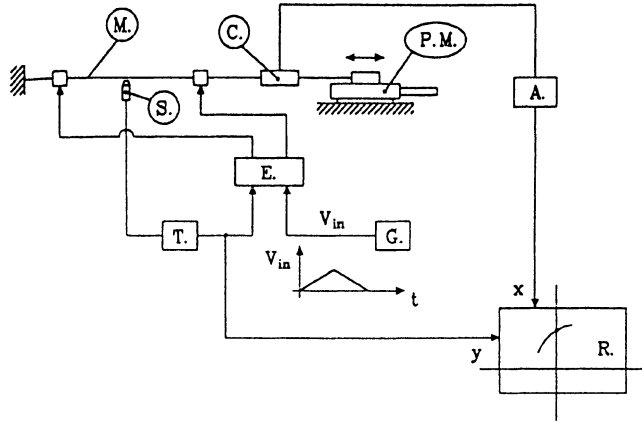


Figure 8: Scheme of the set-up for isometric tests on SMA material

robot hand exert forces without offering motion of the fingers: these are conditions for "isometric" tests. In the isometric test of the SMA material, a determined strain is given to the wire; this strain is maintained for all the thermomechanical cycle in which the force (and consequently stress) of the material, versus varying temperature, is recorded. A scheme of the set-up utilized for isometric test on SMA material is depicted in Fig.8.

On the other hand, during manipulative operation, the actuators of the robot hand (and consequently the active elements) exert forces which are variable with the kinematics configuration of the effector. We have simplified the test conditions by considering constant forces instead of variable loads. During "isotonic" tests, the displacement (strain) of the SMA sample, loaded by a constant force, has been recorded versus temperature variations. The complete characterization of the SMA material has been obtained by considering a sufficient number of isotonic curves, each one correspondent to a determined constant load. By means of isotonic tests it is possible to evaluate the maximum displacements (and related temperatures necessary to obtain them) versus applied load. The scheme of the test-bed for isotonic test is reported in Fig.9.

As a general result from the tests performed on SMA samples, it has been seen that there are intrinsic limitations of the material which maintain its utilization inside a restricted working area. As an example, as it has been recorded by the isometric test, for a given strain it was not possible to augment the temperature of the sample over a specific value of force, without causing irreversible transformations of the material. For these simple

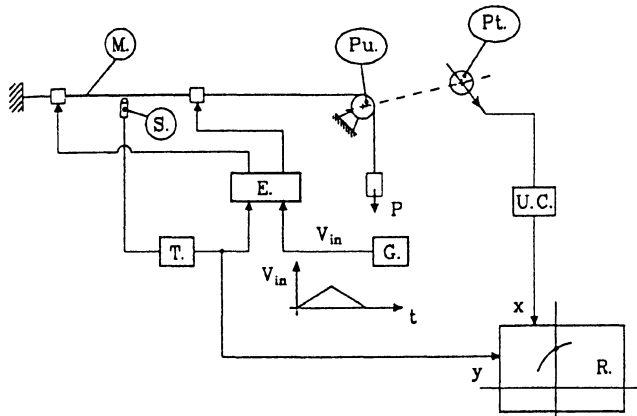


Figure 9: Scheme of the set-up for isotonic tests on SMA material

reasons it is clear that the knowledge of the working area is fundamental when characteristics of actuator repeatability are required. The definition of a working area turns into the definition of a related "limit curve" (Fig.10) (Bergamasco et al,1989) (Bergamasco et al,1990).

From the analysis of the course of isometric curves, obtained from isometric test (Fig.11), it is possible to evict two particular considerations: a) there is a high sensitivity of the stress with respect to temperature; this fact implies that from the technical point of view, a fine control of the force is not possible by a open loop system; it is necessary to realize a closed loop where the controlled variable is the exerted force; b) the presence of hysteresis between the two branches of the curve correspondent to heating and cooling phases introduce a large degree of complexity for the synthesis of the controller. The same considerations apply also for isotonic tests. In Fig.12 a typical curve of isotonic tests is reported.

A further test performed on the SMA material has been the "dynamic" test, consisting in evaluating the frequency respond, at full displacement, of a simple actuator made by a SMA wire connected with a fixed weight. Poor dynamic performances have been obtained (the frequency cut-off was 0,1 Hz); this is perhaps the greater drawback for a practical utilization of SMA materials as robotic actuators. The possibility to overcome the limitations of dynamic performances can be achieved only by introducing complex systems (cooling, protections, etc.) in the design of the actuator system. The next step of the research for the definition of the active element has been that of extending the results obtained on the SMA wires also to other

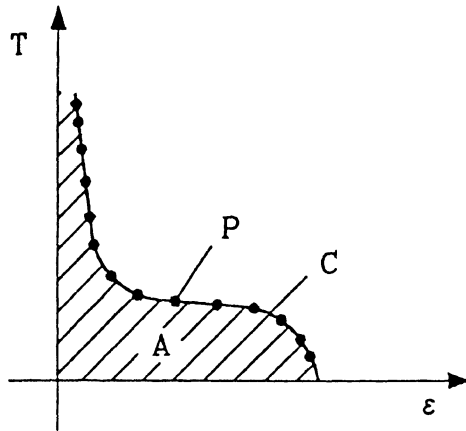


Figure 10: Projections on the plane (T, ϵ) of the limit curve; the shaded region is the correspondent working area; A=working area; P=limit point; C=limit curve.

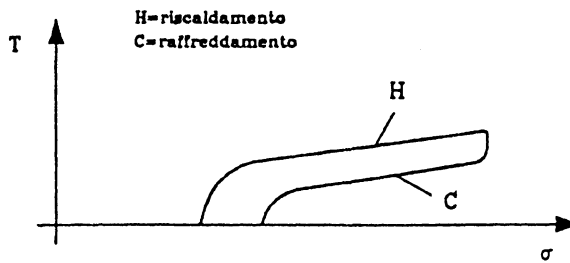


Figure 11: Typical isometric curve

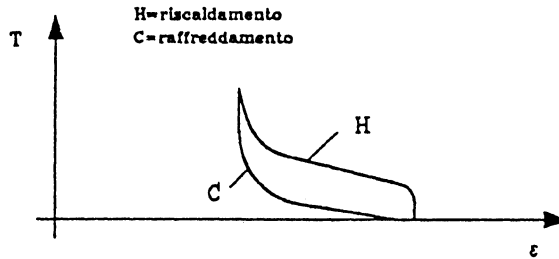


Figure 12: Typical isotonic curve

SMA structures possessing different shapes and different loading conditions. In this framework, when it is required to consider a proper geometry for the final actuators, it is necessary, first of all, to define what kind of mechanical performances the active element must exhibit. If, as explained above, the active element can be approximated to a common structural element, the mechanical performances to be considered are:

- a) maximum stroke in martensitic phase;
- b) maximum force or torque, it can exert, in austenitic phase, without irreversible transformations would appear.

Clearly the mechanical performances of the actuator system will not be the same of those of the active element; this is mainly due to the fact that, in order to obtain a reversible motion at the joint, it is necessary to connect one active element with another (either passive or active) component. In general, depending on the particular mechanical assembly of the actuator system, the mechanical performances of the actuator are only a percentage of those of the active element. The same concepts of “limit curves”, and the correspondent working areas, associated with the SMA material can be translated to the active element.

In fact (Fig.13) the behaviour of the active element is similar to that of the material: in the case of the structural element, however, we have to substitute a generalized force f (force or torque) to the stress, and a generalized displacement s to the strain. A theoretical estimation of the course

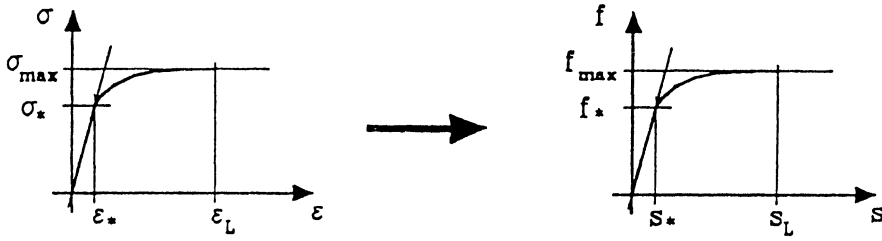


Figure 13: The same concepts of working area can be applied also to the active element

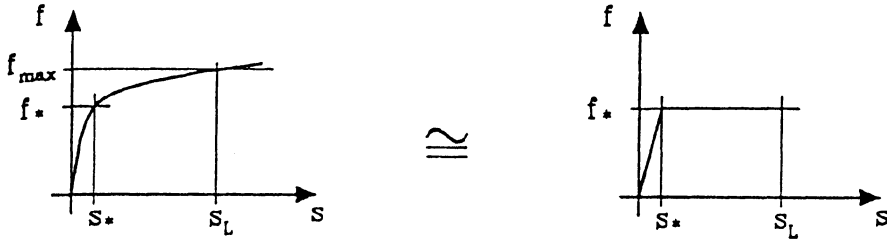


Figure 14: Qualitative estimation of the limit curve for the active element

of the whole limit curve for the active element, starting from experimental data obtained for the material, is a complex task. We have realized that a qualitative estimation of the limit curve (as represented in Fig.14) allows to adequately model the real behaviour of the active element.

Two important values for the determination of the active element behaviour are:

1. f^* : yielding force of the structural element at temperature T^* (the minimum temperature at which the “loss of memory” occurs in conditions of complete austenitic phase);
2. s_L : limit displacement impressed at low temperature (when the ma-

terial is in the martensitic phase) and completely recoverable by the structural element.

In order to evaluate and verify the behaviour of the active element one of the two above values will be considered according to the particular geometry of the identified structural element and to the correspondent loading condition it undergoes. Although the approximated limit curve represents a reduction of the real performances of the active element, it introduces caution condition for design purposes. When loading conditions for the active element have been specified (and consequently also specifications of the mechanical performance of the actuator are known) the choice of the appropriate geometry for the active element must be performed in order to exploit the available volumes in the mechanical structure of the robot link (either a finger phalanx structure or an artificial limb or leg). In this framework SMA are a very powerful technology since they allow, for given mechanical output specifications, different typologies of the active element. We have analyzed both “wire” and “helicoidal spring” geometries. In terms of encumbrances (volumes occupied by the active elements) and for a given mechanical performances, the helicoidal spring possesses lower longitudinal dimensions while wider cross sections with respect to the wire geometry. The final design choice for the appropriate geometry of the active element must take into account also other parameters such as:

- a) total electrical resistance: higher values are desirable when Joule effect is used for the heating of the active element.
- b) thermal exchange: the ratio between the surface (utilized for thermal exchange) and the volume of the active element is fundamental for the definition of the time constant τ regulating thermal transient conditions for a given structural element.

The design of the active element must then take into account, for given output performance, an optimization process between the above different factors aimed at obtaining the lower encumbrances. Usually packaging problems are underestimated during the design of a robotic effector. In this sense SMA technology allows to address the design phase according to an “integrated” approach.

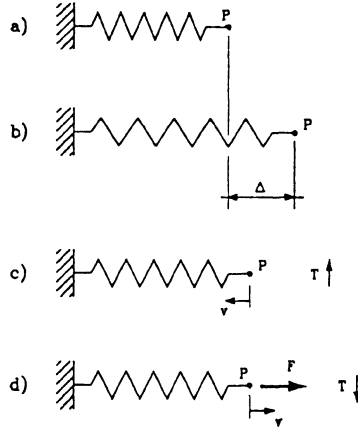


Figure 15: Successive configurations of a SMA helicoidal spring

3 Actuator Design

The analyzed SMA active elements (either the wire or the helicoidal spring) are not capable, per se, of generating a reversible and continuous motion. In fact, as depicted in Fig.15, the heating of the active element will generate a motion (during the recovery of the original shape) in the sense of reducing to zero the elongation of the helicoidal spring.

At this point the geometrical configuration cannot be thermally varied; it is necessary to apply a force F capable of deforming again the structure. From the above consideration it is possible to state that an SMA actuator, intended as a system for performing continuous motion, cannot consist of only one actuator. Among all possible mechanical configurations (a simple passive weight attached to the active element, a passive antagonistic steel spring, etc.) we have chosen the arrangement schematically represented in Fig.16.

It consists of two SMA active elements in series (push-pull configuration). During the assembly of the actuator, the two helicoidal springs are pre-deformed of the same δ with respect to their recovered shape (an optimal δ can be used in order to maximize the stroke of the resulting actuator). By means of the push-pull configuration the motion of the joint can be controlled with the same modalities for both versus of motion and, moreover, the external mechanical characteristics of the resulting actuator will be perfectly symmetrical in both directions. With reference to Fig.16, from the point of active element A, the action of active element B consists in the application of the

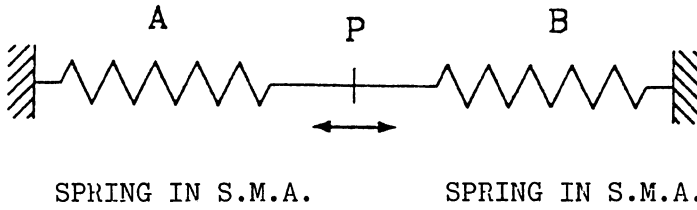


Figure 16: Push-pull configurations of a SMA active elements spring

force F . The push-pull configuration allows to control the force F : in fact, during the heating operation of active element A, F must be very small (ideally zero); if this condition occurs, A will be able to completely recover its shape; in this situation the maximum stroke of the actuator can be reached. The same considerations (with A and B reversed) apply when the joint must rotate or move in the opposite direction. Then, the presence of our active antagonistic active element allows the desired control of the joint. Temperature is the control variable for varying the mechanical characteristics of each active element. The only drawback of the push-pull configuration occurs when the actuator is required to operate at high frequencies. In this situation, in order to obtain the inversion of the joint motion, the heating must be commuted from one active element to the other. If a rapid heating of one active element is not coupled with an equally rapid cooling of the antagonistic element, internal stresses in one or both of the active elements can be induced with the consequent "loss of memory". To avoid this critical problem protection (either mechanical and/or electrical) must be devised. The actual configuration we have used for the development of the SMA actuator is the one represented in Fig.17, where two active elements are connected, through kevlar tendons, to a pulley: the axis of the pulley represents the mechanical output of the SMA actuator.

The heating of the active elements of a push-pull configuration can be performed according to two different modalities: a) alternative (or reciprocating) or, b) complementary heating procedure. By the alternative modality the heating of the active elements cannot be contemporary: in this situation,

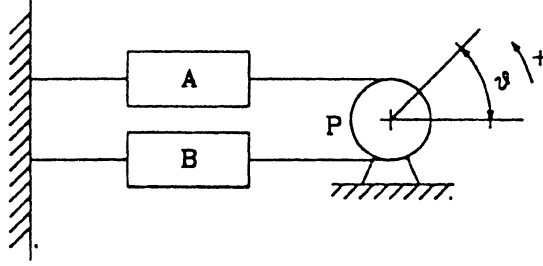


Figure 17: Scheme of the experimental push-pull configuration

at a given time, only one active element (that produces the motion in the desired direction is heated (by Joule effect), while the other is cooled and follows the motion caused by the former. In the complementary modality the two active elements are contemporarily commanded (heated) in such a way that, for example, the sum of the electric powers could be constant. The motion is then generated by acting on the difference of the two electric powers. The complementary modality allows a simpler control, since there is a linear dependence of the angular rotation with the modulator factor of the two powers. This fact could also allow, when a high precision is not required, an open loop control system. Another advantage of the complementary modality is the possibility to augment the tonicity of the system, allowing the actuator to be more robust with respect to mechanical disturbances (variations of the external load). In fact, by referring to Fig.18, every position of the joint is the result of the equilibrium of the three forces T_1 , T_2 , P ; in the case of complementary modality both T_1 and T_2 are controlled. there exist infinite values of T_1 , T_2 , satisfying the equilibrium, differing only by the value of the tone T : $T = \frac{T_1+T_2}{2}$. On the other hand, the stiffness at the axis of the actuator depends on the compliance of the entire elements (when their temperature are maintained constant). But compliances of the two elements depends on the correspondent values of T_1 and T_2 and, after all, from the tone.

The higher is the tone, the higher the stiffness of the active elements must be and, consequently, the higher the compliance of the actuator axis will be (high robustness with respect to mechanical disturbances). As a

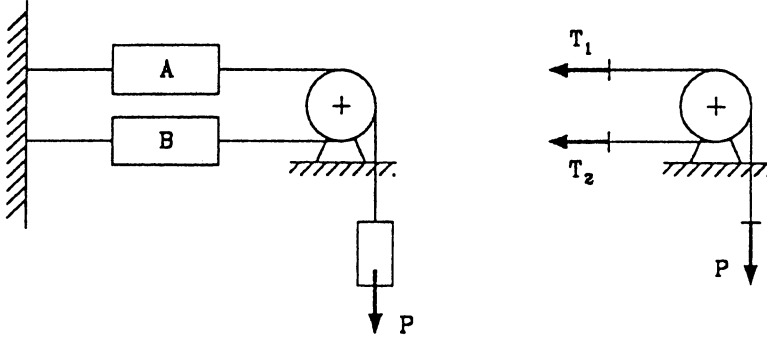


Figure 18: Equilibrium condition for the complementary control modality

consequence, independently from the value of P , the tone associated to the achieved regular position can be varied, in order to allow the system to show different robustness with respect to mechanical disturbances. This cannot be achieved if alternative modality would be used; in fact $T_2 \simeq 0$ and $T_1 \simeq P$, and the tone must be of the same order of magnitude of P . For a small load, the robustness to external disturbances will be low. Alternative control modality, however, shows advantages in terms of frequency cut-off; in fact according to complementary modality, the active elements reach higher temperatures and, consequently the value of frequency cut-off is lower with respect to the alternative modality. Since an acceptable value of frequency cut-off was a fundamental specification for the development of actuators for robot effectors, we have choose, in the course of our work, the alternative control modality. "Alternative" control is characterized by the following factors:

1. the two active elements are never contemporary commanded;
2. the commanded active element is that generates, by its recovering motion, the desired direction of motion.

The logic of the "alternative" control is depicted in Fig.19, where θ_r is the effective angular position, θ_d is the desired angular position and ϵ is the error between θ_d and θ_r . The command is performed to one active element or the other according to the sign of the error.

The problem of the "safety" of the two active elements (the condition of "loss of memory" must never be reached) has been solved by two different

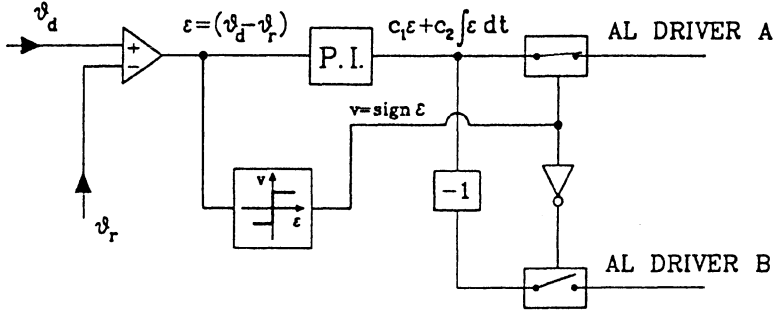


Figure 19: Logic of the alternative control of the SMA actuator

types of "protections": a) thermal, and b) mechanical. The thermal protection consist in inhibiting the electric current supplied to an active element until the temperature of the antagonistic element is greater than a threshold T_0 , appropriately choosen on the limit curve of the active element (Fig.20); if the temperature of the SMA active element is lower than T_0 the damage of the material cannot occur.

The thermal protection, however, presents some practical difficulties since the sensing of temperature in correspondence of the active elements is difficult.

The idea of the mechanical protection takes its origin from the observation that all the protection systems, implemented at the control level, are completely inadequate when the robot effector collides against an obstacle or the external environment. in order to avoid the possibility that such a situation could damage the active elements, we have introduced, by connecting in series with each SMA element a mechanical protection (see Fig.21).

It is realized by a passive mechanical axial coupling (a passive steel spring, in practice) capable of limiting the force acting on the active element. The behaviour of the mechanical protection can be depicted as in Fig.22; perfectly rigid until the tensile force F does not overcome the threshold value $F > F_0$, while perfectly compliant when $F > F_0$. In Fig.22 is represented with respect to the limit curve in the plane T, F , also the consequent restriction of the working area.

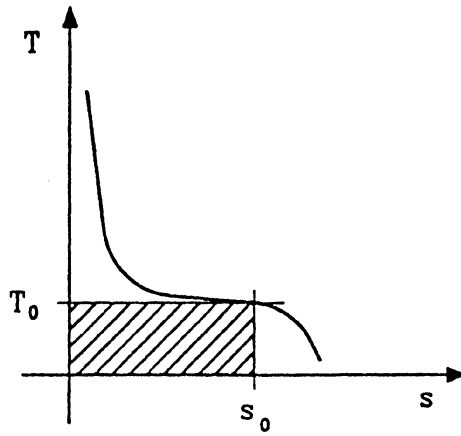


Figure 20: The restricted working area as obtained by using a thermal protection

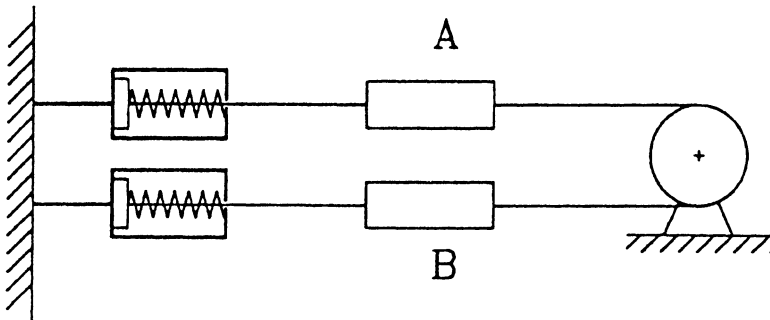


Figure 21: Scheme of the mechanical protections in series with the SMA active elements

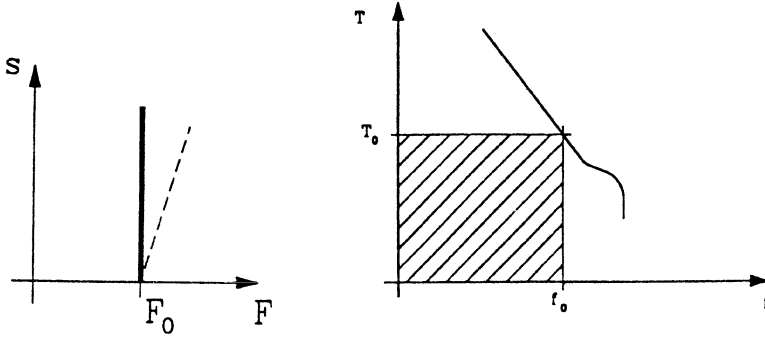


Figure 22: Characteristic of the ideal mechanical protection and the correspondent restricted working area

Since the frequency cut-off of a SMA actuator depends on the rate of thermal transients of the active elements, it has been particularly important to design and realize an appropriate passive cooling system. Although heat sink methods have been already tested and their efficiency proved for wires (Hashimoto, 1985), we have chosen a fluid cooling method. This solution, with respect to the heat sink, easily allows the turning of the removed heat amount especially during the development phase of the work. The adopted fluid has been deionized water performing the cooling of the active element by forced convection. The experimental set-up, completely realized in our laboratory, illustrated in Fig.23, allowed us to verify the performances of the whole actuation system possessing SMA active elements. The experimental set-up consists of: two SMA active elements EA arranged in push-pull configuration; a micrometric table RP allowing the tuning of the pre-loading conditions on the SMA active elements; mechanical protections Pr in series with the active elements; an angular potentiometer P, converted with the pulley Pu, recording and generating the feedback signals to the electronics E (comprehensive of alternative control and power drivers); an hydraulic circuit brings the cooling fluid to the box S where the active elements are positioned.

Dynamic tests have been performed by commanding the system with a sinusoidal input signal (V_{in}) of varying frequency. The resulting frequency plot is reported in Fig.24; the value of the achieved frequency cut-off is $F_m = 1.2Hz$, ten times greater than the one obtained with air cooling method ($F_h = 0,11Hz$).

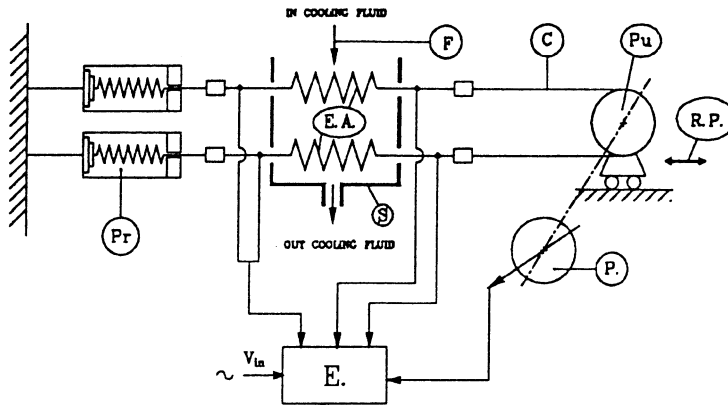


Figure 23: The whole actuation system with SMA active elements

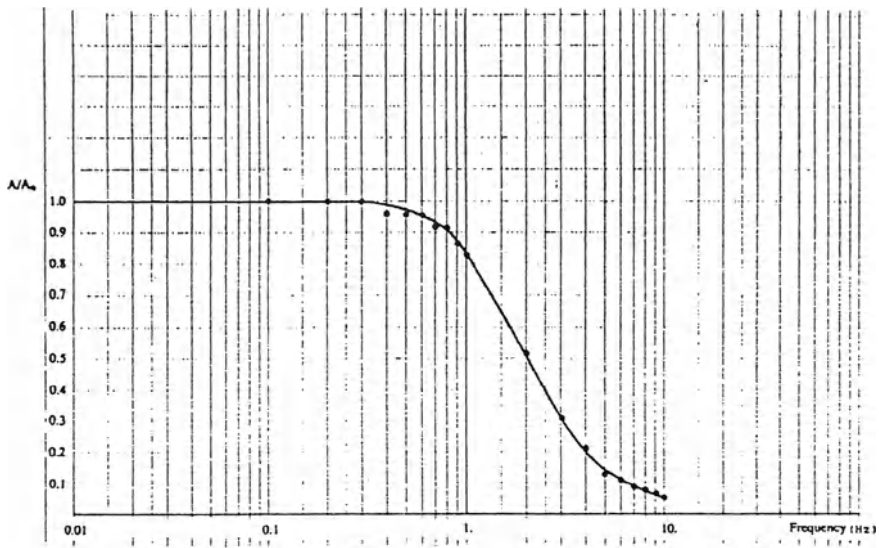


Figure 24: Frequency plot

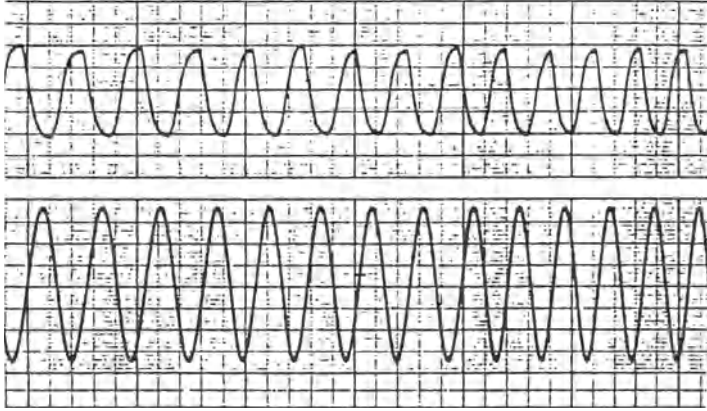


Figure 25: Frequency output (lower part) with an input signal frequency of $0.4Hz$

Fig.25 shows the recording of the output for an input signal frequency of $F = 0,4Hz$.

Besides the dynamic tests, which have shown a satisfying value of the frequency cut-off of the experimental system, the reliability of the actuator has been tested by both considering continuous working cycles for different periods of time and also by blocking the output axis of the actuator as to simulate collision conditions of the robot effector.

4 Conclusions

The described work has collected all the activity we have carried out in order to analyse the different components of a SMA actuator. Each part of the actuation system has been presented by outlining the desired design characteristics and also the effective performances of an experimental prototype that has been realized in our laboratory.

By taking into account the desired mechanical performances, an appropriate choice of the active element geometry could allow to directly integrate SMA actuators inside the structure of each phalanx of a robotic end-effector. In this case it is possible to obtain a sort of direct-drive actuator morphology since the volume of the motor could be located near the joint.

However, the favourable characteristics of SMA material could be exploited also to design actuators possessing a specific shape with appropriate

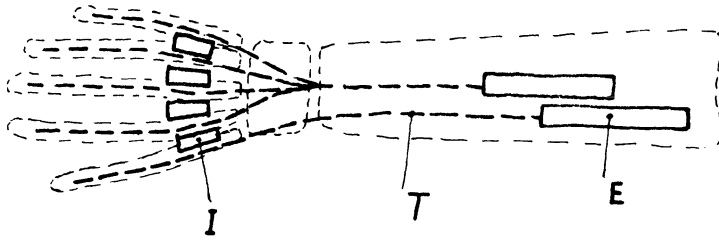


Figure 26: Possible scheme of the arrangement of extrinsic and intrinsic actuators for future anthropomorphic robot hands; I=intrinsic actuators; E=extrinsic actuators; T=tendons.

mechanical performances. If we consider, for example, the possibility of developing an actuating system possessing real anthropomorphic features, SMA active elements geometry could be designed in order to fulfil the required specifications in terms of occupied volumes with respect to the desired mechanical output.

In this framework we could imagine to design both "extrinsic" and "intrinsic" actuators: the former possessing strong mechanical performances, especially in terms of generated force, while the second showing higher position accuracy and lower strength. The simple geometry of SMA wires could implement "extrinsic" actuators since, as realized also by (Nakano et al.,1984) in the Hitachi's hand, the forearm structure can envelope the whole actuation package. On the other hand, helicoidal springs as intrinsic actuators could generate the desired controllability of small forces with large displacements.

Furthermore, if we move our attention from the geometry of the single actuator to the arrangement of the whole actuation system for the design of future anthropomorphic robot hands, it is possible to devise a biomorphic disposition of the actuators, as the one represented in Fig.26, that could replicate, at least from the morphological point of view, the arrangement of tendons and muscles of the human hand.

SMA technology could allow to implement such a complex system, but more research activity must be done in order to solve the problem also from the functional point of view.

In the presented experimental work on SMA actuators we have seen that SMA technology, with the appropriate design of the whole system, represents a reliable solution, also in terms of fatigue. The protection systems that we have introduced can assure the operation of the SMA actuator also under heavy operating conditions. However the acquisition of a higher frequency cut-off can be obtained only by introducing complex technical solutions for the cooling system.

Experimental results are very encouraging: now a large effort is required for refining and evaluating the proposed actuator design.

References

Bergamasco M., Salsedo F., Dario P. (1989),: *A linear SMA motor as direct-drive robotic actuator*, Proc. of IEEE Intern. Conf. on Robotics and Automation, Scottsdale, AZ.

Bergamasco M., Dario P., Salsedo F. (1990),: *Shape Memory Alloy Microactuators*, Sensors and Actuators, A21-A23.

Gabriel K.J., Trimmer W.S.N., Walker J.A. (1988),: *A micro rotary actuator using shape memory alloys*, Sensors and Actuators, 15.

Hashimoto M., Takeda M., Sagawa H., Chiba I., Sato K. (1985),: *Application of shape memory alloy to robotic actuators*, Journal of Robotic Systems, 2(1).

Hirose S., Ikuta K., Umetani Y. (1984),: *A new design method of servoactuators based on the shape memory effect*, Proc. of RoManSy '84.

Hirose S., Ikuta K., Tsukamoto M., Sato K. (1987),: *Considerations on design of the actuator based on the shape memory effect*, Proc. of the 7th IFToMM World Congress, Sevilla, Spain.

Ikuta K., Tsukamoto M., Hirose S. (1988),: *Shape memory alloy servo actuator system with electric resistance feedback and application for active endoscope*, Proc. of the 1988 IEEE Intern. Conf on Robotics and Automation, Philadelphia, PA.

Kuribayashi K. (1986),: *A new actuator of a joint mechanism using TiNi alloy wire*, Inter. J. Robotics Res., Vol.4, No.4.

Nakano Y., Fuje M., Hosada Y. (1984),: *Hitachi's robot hand*, Robotics Age, July.

Otsuka K., Shimizu K. (1986),: *Pseudoelasticity and shape memory effects in alloys*, Intern. Metal Reviews, Vol.31, No.3.

Perkins J., Edwards G.R., Such C.R., Johnson J.M., Allen R.R. (1975),: *Thermomechanical characteristics of alloys exhibiting martensitic thermoelasticity*, Proc. of Shape Memory Effect in Alloys, Plenum, New York.

Yaeger J.R. (1984),: *A practical shape-memory electromechanical actuator*, Mechanical Engineering, Vol.106, No.7.

CCD Retina and Neural Net Processor*

A. M. Chiang

Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02173

1. Introduction

In a charge-coupled device (CCD), a continuous analog signal can be sampled and represented as analog charge packets. The charge packets can then be stored, delayed, transferred, reproduced, divided and summed [1]. A charge-domain, high-speed, low-power multiplier has been developed based on these fundamental CCD properties [2]. In addition, a charge domain parallel processing architecture has been developed [3] that allows single-chip CCDs to perform more than one billion computations per second and dissipate less than 2W [4],[5]. Because of their inherent and developed capabilities, CCDs are ideally suited for hardware implementation of neural nets with a large number of neurons and interconnections. Two types of CCDs will be described: CCD solid state retinas for early vision processing such as image restoration and segmentation, and CCD two-layer neural net (NN) processors for signal processing applications such as image feature extraction and pattern classification.

2. CCD Retina

The integration of signal processors with photosensors makes it possible to perform simultaneous, parallel computations on each pixel in real time (Fig. 1). An interline-transfer area imager can be designed with integrated CCD signal-processing elements between neighboring pixels to simulate a solid state or a biological system with locally connected interactions between its neighboring cells. In particular, a retina can be designed to perform computations such as time derivatives of time-varying pixel intensity function and minimization of the glo-

* This work was sponsored by the Department of the Air Force and the Office of Naval Research

bal energy function due to local spatial intensity interactions. The retina can be used for image restoration of a corrupted, noisy optical input, and also for motion detection.

2.1 Device Concept

A solid state retina capable of restoring and segmenting an noisy image is described here. Given a measured image, assuming the measurement is a corrupted or a distorted version of the original image, the purpose of this device is to reconstruct or restore the original, uncorrupted scene by estimating its *maximum a posteriori* (MAP) distribution. In recent years, many researchers have tried to solve this optimization problem for early vision processing. But, so far, most efforts have been concentrated in computer vision using a stochastic approach such as simulated annealing [6] to solve this problem [7],[8],[9]. Only recently, a deterministic approach using a set of dynamic parallel update equations has been introduced, which is more suited for VLSI implementation [10],[11]. The retina to be described is based on the second approach.

In general, intensities of neighboring pixels in an image are correlated: the probability function for the intensity of a given pixel is fully specified by the intensity values of its neighboring pixels. This image attribute can be modeled by a local pixel-intensity probability distribution function such as Markov random fields (MRF):

$$p(I_{ij}|I_{kl}, kl \neq ij) = p(I_{ij}|I_{kl}, kl \in N_{ij}) \quad (1)$$

where I_{ij} is the intensity and N_{ij} is the neighborhood of the pixel ij . Furthermore, the Hammersley-Clifford theorem can be used to prove that the pixel intensity of an image is a MRF if and only if its joint distribution function is a Gibbs distribution [7]:

$$p(I) \propto \exp[-\sum_{ij} U(I_{ij})] \quad (2)$$

where

$$U(I_{ij}) = \sum_{kl} u(I_{ij}, I_{kl}) \quad \text{for } kl \in N_{ij},$$

and u is defined as a local intensity interaction function between neighboring pixels. Images generated by sensors such as video cameras or radar sometimes are distorted or contain a noise term introduced by the measurement process. These measured images can be modeled by

$$p(I^m|I) = \exp \sum_{ij} -b(I_{ij}^m - I_{ij})^2 \quad (3)$$

where I_{ij}^m is the measured intensity of pixel ij . Bayes' theorem allows the *a posteriori* probability to be represented as

$$p(I|I^m) = \frac{p(I^m|I)p(I)}{p(I^m)} \quad (4)$$

Substituting Eqs. (2) and (3) into (4), we obtain

$$p(I|I^m) = \frac{e^{-E}}{Z} \quad (5)$$

where Z is a normalization constant. We can see that the *posteriori* distribution function can also be expressed as a Gibbs function with an equivalent energy function E defined as

$$E = \sum_{(ij)(kl)} u(I_{ij}, I_{kl}) + \sum_{(ij)} b(I_{ij}^m - I_{ij})^2 \quad (6)$$

It becomes clear now that we can estimate the original noiseless image by computing the MAP distribution $p(I|I_m)$. In practice, to solve this MAP problem, it is more convenient to work directly with the energy function and to compute the minimum energy state rather than maximize the *a posteriori* distribution function.

It should be noted for a 128×128 -pixel 256-gray level image, the total number of possible energy states is 256^{16384} . To find a minimum from all the possible states is a formidable task. By means of the Metropolis algorithm [12], the minimum energy state can be estimated by simulated annealing or stochastic relaxation. Using this approach in [7], Geman and Geman have demonstrated good image restoration and segmentation results. For VLSI implementations, it is preferable to use a gradient descent method to minimize the energy function, which allows us to derive a set of simultaneous dynamic equations for updating pixel intensities. Define a force function at each pixel by

$$f_{ij} = \frac{\partial E}{\partial I_{ij}} \quad (7)$$

A minimum energy state can be estimated by using an iterative method simultaneously updating each pixel intensity in the following way:

$$I_{ij}^e(n+1) = I_{ij}^e(n) - wf_{ij}^e \quad (8)$$

where I_{ij}^e represents estimated original intensity of pixel ij , n represents the number of iterations and w controls the rate of change of intensities of a given pixel.

2.2 Retina with a Local Quadratic Interaction Function

In this section, a solid state retina with a nearest-neighbor quadratic interaction function will be described. From (2), the energy function between neighboring pixels can be written as

$$\sum_{kl} u(I_{ij}, I_{kl}) = a[(I_{ij} - I_{ij+1})^2 + (I_{ij} - I_{ij-1})^2 + (I_{ij} - I_{i-1,j})^2 + (I_{ij} - I_{i+1,j})^2] \quad (9)$$

Each pixel (photodiode) on this retina would be connected to its nearest neighbors with the interconnection strength proportional to the gradient of a pair-interaction function shown in Eq. (9). Substituting (2),(6),(7),(9) into (8), we have a set of dynamic parallel pixel-intensity update equations

$$I_{ij}^e(n+1) = I_{ij}^e(n) - w[b[I_{ij}^e(n) - I_{ij}^m] + h(I_{ij}^e(n), I_{ij+1}^e(n), I_{ij-1}^e(n), I_{i-1,j}^e(n), I_{i+1,j}^e(n))] \quad (10)$$

where

$$h(I_{ij}^e, I_{ij+1}^e, I_{ij-1}^e, I_{i+1,j}^e, I_{i-1,j}^e) \equiv 2a[(I_{ij}^e - I_{ij+1}^e) + (I_{ij}^e - I_{ij-1}^e) + (I_{ij}^e - I_{i+1,j}^e) + (I_{ij}^e - I_{i-1,j}^e)]$$

2.2.1 Device Design

A layout diagram of a CCD retina with a nearest-neighbor quadratic interaction function is shown in Fig. 2. Each pixel is connected to its four nearest neighbors, and the photodiode at each pixel is separated into two halves with a CCD readout line shift register running through the center of the pixel. The connection strength to each neighbor is equal to the gradient of the interaction function, in this case a linear connection strength. As can be seen in Fig. 2, the connection between pixels ij and $ij+1$ is implemented by using a pair of CCDs, with the output of one CCD connected to pixel ij and of the other to pixel $ij+1$. A floating gate is used to sense the photovoltage generated on the n^+ side of the photodiode. The usual "fill and spill" inputs are modified so that for the CCD connected to pixel ij , the input signal voltage is the photovoltage V_{ij} on pixel ij and the "reference" voltage is the photovoltage on pixel $ij+1$, and for the CCD connected to pixel $ij+1$, vice versa. When the common input diode is then pulsed on, negative charge proportional to the difference between the photovoltages of pixels ij and $ij+1$ is injected into the CCD connected to the pixel with the larger photovoltage, but no charge is injected into the other CCD. After the injected charge is transferred out of the CCD, the difference in photovoltage between the two pixels is reduced. It is to be noted that this operation is carried out simultaneously on all the pixels. It follows then the system is left in a lower energy or more probable state. The CCD structure shown in Fig. 2 is the basic building block for a smart retina. With simple modification, a nonlinear coupling function can be implemented. One such example will be described in the next section.

The readout organization of the retina is based on an interline-transfer CCD imager where the signal generated from each pixel is transferred to a vertical CCD linear shift register and all the vertical lines are clocked in parallel. This readout scheme is chosen because it provides high quantum efficiency at the sensors, lower noise readout than the conventional xy-scanner, and the possibility of integrating anti-blooming control at each sensor for large dynamic range applications.

For motion-detection purposes, the retina will be designed to have two CCD shift register stages at each pixel site rather than one, so that two consecutive time samples can be taken at each pixel. The time difference between the two samples will be programmable and controllable by the CCD clock rate and can be as fast as 50-100 ns. The device will use a horizontal CCD shift register to read out the image on a line-by-line basis. At the output diffusion node, a correlated double sampler takes the difference between two time samples and performs the motion detection function on each pixel in real time as it is read out.

2.3 Retina with a Lorentzian Interaction Function

The interaction function shown in Eq. (3) can be chosen to correspond to the MRF of any typical image. As shown in previous section, a simple Gaussian distribution which corresponds to a local quadratic interaction function was used to model an image. The problem with this approach is that it does not preserve image boundaries. The retina may evolve to an undesirable minimum energy state in which all neighboring pixels have the same intensity value. For image restoration applications, a retina with a Lorentzian interaction function would yield better results [10]. The Lorentzian function is given as

$$\sum_k \mu(I_{ij}, I_{kl}) = g(I_{ij}, I_{ij+1}) + g(I_{ij}, I_{ij-1}) + g(I_{ij}, I_{i+1,j}) + g(I_{ij}, I_{i-1,j}) \quad (11)$$

where

$$g(I_{ij}, I_{ij+1}) \equiv \frac{1}{1 + (I_{ij} - I_{ij+1})^2}$$

A CCD retina with connection strength corresponding to the gradient of this Lorentzian function can be implemented by using four CCD units described above. A block diagram of this device is shown in Fig. 3.

3. CCD Neural Net Device

A single-chip NN processor that can be used to compute the matching score between two layers of neural nodes, and to select and enhance the maximum output, is described here [13]. This type of NN device can be used for image feature extraction or pattern classification. The input layer consists of N analog nodes and the output layer of M analog nodes. There are two types of connectivity between the two layers: one is a fully connected (FC) net, in which every input node is connected to every output node via a variable connection weight; the other is a sparsely connected (SC) net, in which only a limited number of the input nodes are connected to each output node. The two nets share a lot of common design features: the

difference is that they are intended for different applications. The SCNN is developed specifically for feature extraction applications. At each of the output nodes of either of these nets, the weighted sum of the corresponding inputs is computed and a sigmoid type nonlinear detection function is applied to the weighted sum. A decision is then made on the basis of the M detected outputs. For example, a FCNN device with 400 input nodes and 30 output nodes would provide 12,000 programmable connections in an estimated chip area of 1.2 cm^2 . Clocked at 10 MHz, the chip performs 4 billion connections/sec, clearly demonstrating the connection density and throughput rate advantages offered by the CCD technology.

These CCD NN devices are generic connection modules with analog tapped-delay lines as input buffers and on-chip memory for storing digital weights. The on-chip analog input buffer enables the CCD modules to implement hierarchical layered networks in a high-throughput pipelined fashion without massive I/O interconnects between modules. By using two such CCD modules and feeding the output of the first module into the second one, the two-chip set can be used to implement a three-layer net, and a three-chip set can be used to implement a four-layer net, and so forth. In addition, systems with larger interconnection between adjacent layers can be realized by using many CCD modules in a parallel fashion. Again, the input buffer minimizes the I/O problem usually associated with systems having a large number of ICs.

The design considerations and applications for the two CCD NN devices will be described next. In addition, two charge-domain peripheral circuits, a sigmoid detection and a maximum (winner-take-all) detection circuit will also be presented.

3.1 Fully Connected Neural Net Device

3.1.1 Device Architecture

The CCD structure shown in Fig. 4 is a building block FCNN processor with N inputs and one output. The processor can be used to perform the most common NN computation, the weighted sum of N analog inputs. The weights are programmable. As can be seen in Fig. 4, the processor consists of N -stage analog floating-gate tapped delay lines for shifting and holding the N analog inputs, N CCD multiplying D-to-A converters (MDACs) [2], and an N -stage digital memory for shifting and holding the digital connection weights. At each stage of the CCD shift register, the floating gate is coupled to the analog input port of the corresponding MDAC, and the output from each MDAC is a charge packet proportional to the product of the analog input and the digital weight. All the MDACs share a common output, and the outputs from all the MDACs are summed together in the charge domain.

A single-chip FCNN processor with N inputs and M outputs is drawn schematically in Fig. 5. Every input is connected to every output node via a programmable connection weight. In addition, in many existing NN algorithms, it is important to incorporate a nonlinear function at the output node. The nonlinear function can be modeled as a sigmoid-type function with an offset, which can be implemented straightforwardly in the charge domain by a one-stage bucket-brigade device (BBD). The properties of this one-stage BBD device will be described in Sec. 4.1.

A block diagram of the CCD NN processor is also shown in Fig. 5. The chip consists of N -stage floating-gate tapped delay lines for shifting and holding the N analog input values, N CCD MDACs and $N \times M$ -stage CCD digital memories for shifting and holding the digital connection weights. The layout of this processor is very similar to the single-node processor described in the previous section, except in this case, for each MDAC, there is a M -word bit-parallel CCD digital memory to store the M connection weights. The M words are sequentially addressable.

The device operates as follows: After the M analog inputs x_i and the $M \times N$ weights w_{ij} are loaded into the processor, each word of the N M -word bit-parallel CCD digital memory will be sequentially applied to the MDACs. After the first clock period, the total summed charge from all the MDACs is $\sum_{i=0}^{N-1} w_{i0}x_i$, i.e., the device computes the first output of the second layer,

u_0 . The procedure continues until after M clock periods, the summed charge is $\sum_{i=0}^{N-1} w_{iM-1}x_i$,

that is, u_{M-1} . The maximum of these values can then be selected and enhanced. The connection weights corresponding to the maximum output node can be adapted according to learning rules. A charge-domain maximum detection circuit will be described in Sec. 4.2.

3.1.2 Application for the CCD FCNN Device

As mentioned previously, the CCD FCNN device is a generic weighted-sum processor, which can be used to implement or emulate a variety of neural network algorithms. An example is given here to demonstrate the flexibility and modularity of the device.

The FCNN device described above can be used to implement Sejnowski and Rosenberg's NETalk [14]. NETalk is a neural network that learns to convert written English text into a phonetic representation; it learns to generate speech from text. The network architecture consists of three layers. The input layer has 203 nodes, the hidden layer 60 nodes, and the output layer 26 nodes. The nodes in the hidden layer are fully connected to both the input and output layer. Given a CCD FCNN device with 208 inputs and 30 outputs (the estimated chip area for this FCNN is only 0.72 cm^2), only three such CCD FCNN devices are needed to

implement the NETtalk architecture with 13,740 connections. Since the hidden layer has 60 nodes and each FCNN device only has 30 output nodes, two CCD FCNN chips are used in parallel to implement the fully connected input-to-hidden-layer net. The outputs from the two NN chips can be read into the third chip which is used to perform the connection between the hidden to output layer. Clocked at 10 MHz, once the input is loaded into the NN chips, the CCD chip set would finish one forward pass in 6 μ s.

3.2 CCD Sparsely Connected Neural Net Processor

3.2.1 Device Architecture

A block diagram of the CCD SCNN processor is shown in Fig. 6. The major difference between this device and the FCNN device is that for a net with N input nodes and M output nodes, there are only KM connections between the two layers where K is much smaller than N . In addition, this processing device has an on-chip continuous CCD analog line-buffer memory that allows the device to extract feature information with a moving 2-D template. For illustrative purposes, a SCNN device consisting of a CCD floating-gate tapped delay line for shifting and holding 7 input lines of a gray-level image, 49 MDACs and 20 7×7 CCD digital shift-register memories is shown in Fig. 6. The 7-line buffer memory is connected in series using a serpentine organization. The 7×7 digital memories are for storing feature patterns. Each feature can be a gray-level image with 8-bit accuracy in pixel intensity. At the first 7 cells of every line memory, floating-gate taps are used to nondestructively sense the pixel value and each tapped output is coupled to its corresponding CCD MDAC. (For simplicity, the taps and MDACs for only the first and the last line memory are shown in Fig. 6.)

The operation of using this SCNN device for feature extraction is as follows: A 2-D image is continuously loaded into the processor line-by-line fashion. After a latency time during which the first 6 lines and the first 6 pixel elements of line 7 of an image are loaded into the device, at each of the following load-data clocks, inner products of the input image and the weighting patterns are sequentially computed and feature information is extracted. This has the effect of scanning a 7×7 window through the first seven lines of an input image. It follows then after the first 6 elements of line 8 are loaded into the device, at the following load-data clocks, the device is ready to compute the inner products of lines 2 to 8 of the input image and the 7×7 template. The procedure continues until the inner products between the templates and the last 7 lines of the image are computed and the complete feature information of the input image is extracted. Once a CCD SCNN device with a 7×128 pixel delay line is built, the device can create 20 different 7×7 -template feature maps from a 128×128 -pixel input image in every 32 ms. The estimated chip area of this SCNN device is 0.5 cm^2 .

3.2.1 Application

The SCNN processor is ideally suited for implementing a space-shift-invariant model for an image classifier such as a neocognitron network[15]. The neocognitron is a multilayered network with a hierarchical structure similar to the hierarchical model for the visual system proposed by Hubel and Wiesel [16]. The structure consists of a cascaded connection of many layers of cells. The information of a stimulus pattern given to the input layer is processed step by step in each stage of the multilayered network. A cell in a higher layer responds selectively to a more complicated feature, and is less sensitive to shifts in position of the stimulus patterns than that of a cell in a lower layer. The CCD SCNN processor can be used to implement one layer of the neocognitron model and extract feature information of a given layer. The multilayer system can be implemented by using several CCD processors in a pipeline form.

4. Peripheral Circuits

4.1 Sigmoid Detection Circuit

In general, computational elements or nodes used in neural net models are nonlinear. The simplest node sums weighted inputs and passes the result through a nonlinear detection circuit. One commonly used nonlinear function is a sigmoidal nonlinearity. A simple, low-power charge-domain implementation of this function is proposed here. The charge transfer efficiency from an n^+ diffusion region over a bias gate to a receiving well as a function of device parameters, clock frequency and sinusoidal clock waveform has been solved by Thornber [17]. Based on his derivation, a nonlinear charge detection circuit consisting of a one-stage bucket-brigade device (BBD) is proposed here. The device has a programmable threshold level; the amount of charge transferred through the BBD is controllable by the amplitude of the transfer gate. The nonlinear dependence of the output charge on the input charge of this BBD for two different threshold levels is shown in Fig. 7. One can see that if the incoming weighted-sum signal charge is below a given threshold level, no output charge will be transferred to the output port; therefore, the incoming stimuli will be ignored. On the other hand, if the weighted-sum signal charge is above the threshold level, the amount of charge transferred to the output port is a sigmoid-type function of the difference of the weighted-sum charge input and the threshold level.

4.2 Winner-Take-All Detection Circuit

One of the important features of existing neural network algorithms is the competitive learning mechanism [18]. Competitive learning takes place in a context of sets of hierarchically layered networks. Each layer consists of a set of clusters of mutually inhibitory nodes. The nodes within a cluster inhibit one another in such a way that only one node per cluster may be active: a winner-take-all mechanism. Networks learn to encode the input based on this mechanism. Therefore, in a cluster with M nodes, there is a need to develop a circuit which can detect the location and magnitude of a node with a maximum output value. An analog winner-take-all circuit has been reported recently, but the circuit only selects the maximum output quantity without knowing which node has the maximum value [19]. A low-power charge-domain unit which can select and locate the largest of M analog charge packets is proposed and described here. In a serial mode, the M charge packets are loaded into the device sequentially. After each clock cycle, the unit examines two of the charge packets, retains the larger one and discards the other; a buffer and a digital counter keep track of the location of the retained charge. After M clock cycles, only the maximum charge packet remains in this unit and the buffer retains the address of this charge packet. In a parallel mode, approximately M such units would be used in a pipelined, binary-tree fashion to select and identify the maximum charge packet.

The basic unit, shown in Fig. 8, consists of a CCD two-stage tapped delay line and a flip-flop. Each tapped output is connected to an input of the flip-flop. The flip-flop outputs are used to drive the transfer gates shown in Fig. 8. In addition, the output node A is used to update the address of the maximum charge packet. The circuit operates as follows: Two packets of charge Q_1 and Q_2 are loaded into the tapped delay line under gates G_a and G_b , respectively. If Q_1 is larger than Q_2 , node B of the flip-flop will be high, the transfer gate B will be on and the transfer gate A off, so Q_1 will stay in the delay line and Q_2 will be discarded. Alternatively, if Q_1 is smaller than Q_2 , the transfer gate A will be on and the gate B will be off. It follows that Q_2 will be retained and Q_1 discarded. In a serial mode, the retained larger charge packet will then be shifted under G_a and a new charge packet Q_3 loaded under G_b . The whole procedure repeats until the M th charge packet passes through the device. The update between the value in the digital counter and the selected location of the maximum charge packet is straightforward. Whenever the flip-flop output node A signals a new charge packet is greater than the currently stored maximum, the number in the counter will update a buffer which saves the address of the maximum charge packet. So the circuit not only selects the largest charge packet, but also finds its location.

5. Summary and Discussion

A CCD retina concept that integrates signal processors with photosensors on a chip has been described. A particular design that can be used for image restoration of a corrupted, noisy optical input, and also for motion detection has been given. The basic design concept can be extended to implement an on-center-off-surround, or an off-center-on-surround retina for edge detection. Next-nearest-neighbor or even higher-order interactions can be incorporated on a chip, allowing us to build retinas for image segmentation or feature detection.

The CCD architecture for a two-layer neural net processor for either a fully connected or a sparsely connected implementation has been described. The architecture is very flexible. For example, for a CCD FCNN device with 400 input nodes and 30 output nodes, the processor will provide 12,000 programmable connections. With minor modification at the device output port, the processor can be reconfigured into a two-layer net with 100 input nodes and 120 output nodes. The device is capable of performing 4 billion computations per second with a 10 MHz clock rate. Such performance clearly demonstrates the adaptability and the computation power offered by CCD technology.

References

- [1] A. Gersho, "Charge-Transfer Filtering," *Proc. IEEE*, vol. 67, pp. 196-218, 1979.
- [2] A. M. Chiang and B. E. Burke, "A High Speed CCD Digitally Programmable Transversal Filter," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 745-753, 1984.
- [3] A. M. Chiang, "A New CCD Parallel Processing Architecture," in *VLSI Systems and Computations*, H. T. Kung, Ed., Computer Science Press, pp. 408-418, 1981.
- [4] A. M. Chiang, P. C. Bennett, B. B. Kosicki, R. W. Mountain, G. A. Lincoln and J. H. Reinold, "A 100-ns CCD 16-Point Cosine Transform Processor," *ISSCC-87 Tech. Digest*, pp. 306-307, February 1987.
- [5] A. M. Chiang, "A CCD Parallel Pulse-Doppler Radar Processor," *GOMAC Digest*, pp. 505-509, 1986.
- [6] S. Kirkpatrick, C. Gellatt and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983.
- [7] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721-741, 1984.
- [8] A. Blake and A. Zisserman, "Visual Reconstruction," MIT Press, Cambridge, MA, 1987.
- [9] J. Marroquin, S. Mitter and T. Poggio, "Probabilistic Solution of Ill-Posed Problems in Computational Vision," *Proceedings Image Understanding Workshop*, McLean, VA, pp.

- 293-309, 1985.
- [10] T. G. Vold, "Bayesian Image Processing, Gibbs Distributions, and Neural Networks," *IEEE International Conf. on Neural Nets*, June 1987.
 - [11] D. Geiger and F. Girosi, "A Visual Model for Smoothing, Enhancing and Detecting Discontinuities," preprint A.I. Memo, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.
 - [12] N. Metropolis, A. Rosenbluth, M. Rosenbluth and A. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Phys. Chem.*, vol. 21, pp.1087-1091, 1953.
 - [13] R. P. Lippman, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, pp. 4-22, April 1987.
 - [14] T. J. Sejnowski and C. R. Rosenberg, "Parallel Networks That Learn to Pronounce English Text," *Complex System*, vol. 1, pp. 145-168, 1987.
 - [15] K. Fukushima, S. Miyake and T. Ito, "Neocognitron: A Neural Network Model for a Mechanism of Pattern Recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 5, pp. 826-834, 1983.
 - [16] D. Hubel and T. Wiesel, "Sequence Regularity and Geometry of Orientation Columns in the Monkey Striate Cortex," *J. Comp. Neur.*, vol. 158, pp. 267-294, 1974
 - [17] K. K. Thornber, "Incomplete Charge Transfer in IGFET Bucket-Brigade Shift Registers," *IEEE Trans. Electron Devices*, vol. ED-18, pp. 941-950, 1971.
 - [18] J. L. McClelland and D. E. Rumelhart, "An Interactive Activation Model for Context Effects in Letter Perception," *Psychological Review*, vol. 88, pp. 375-407, 1981.
 - [19] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, C. A. Mead, "Winner-Take-All Networks of $O(N)$ Complexity," Conference on Neural Information Processing System-Natural and Synthetic, Denver, Nov. 28-Dec. 1, 1988.

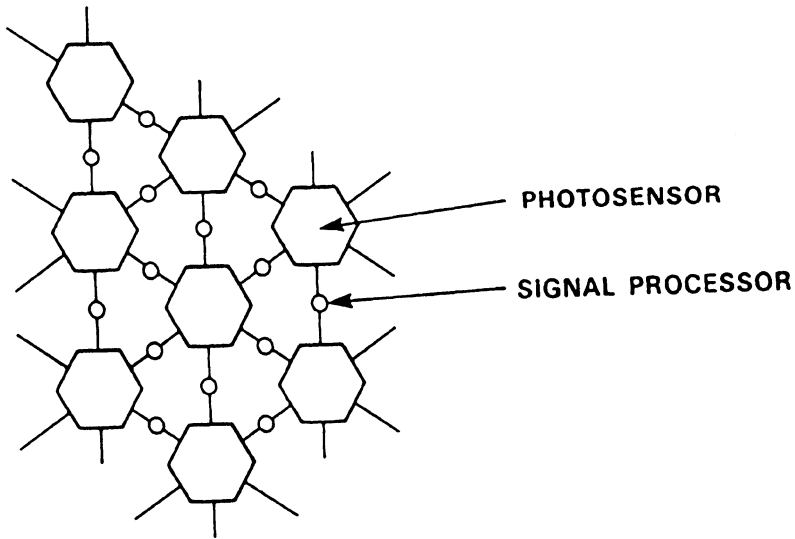


Fig. 1 Schematic of a CCD solid state retina

$$u(l_{ij}) = a[(l_{ij}-l_{ij+1})^2 + (l_{ij}-l_{ij-1})^2 + (l_{ij}-l_{i-1,j})^2 + (l_{ij}-l_{i+1,j})^2]$$

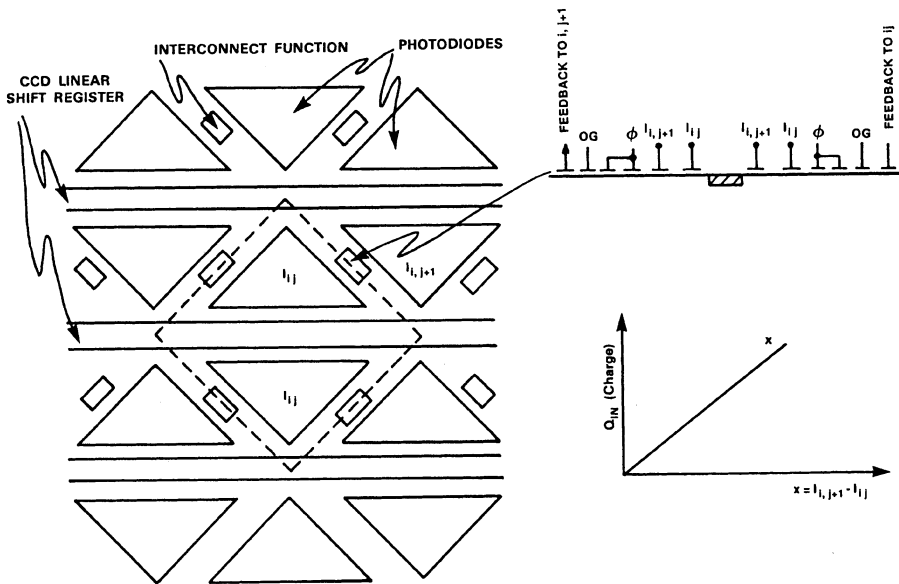


Fig. 2 Layout diagram of a CCD retina. Each pixel interacts with its four nearest neighbors with a quadratic function

$$u(l_{ij}) = g(l_{ij}, l_{ij+1}) + g(l_{ij}, l_{ij-1}) + g(l_{ij}, l_{i+1,j}) + g(l_{ij}, l_{i-1,j}) \quad \text{where} \quad g(l_{ij}, l_{ij+1}) \equiv \frac{1}{1 + (l_{ij} - l_{ij+1})^2}$$

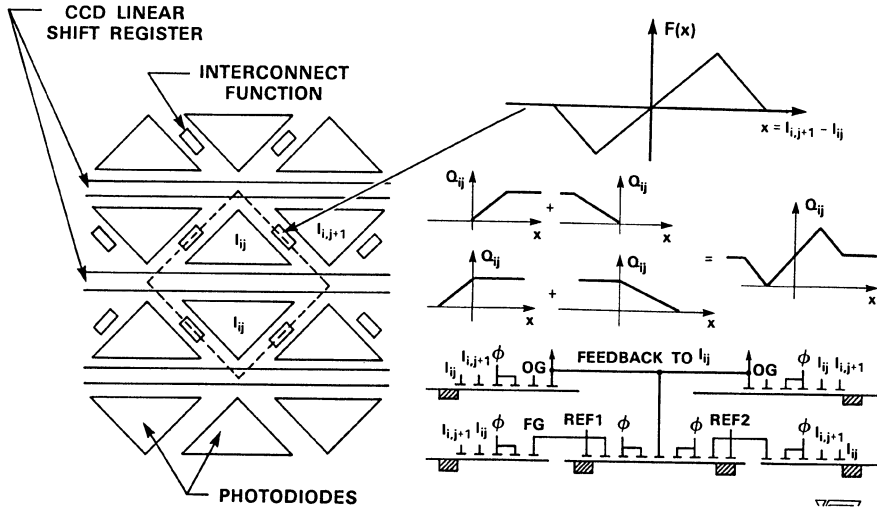


Fig. 3 CCD retina with a local Lorentzian interaction function

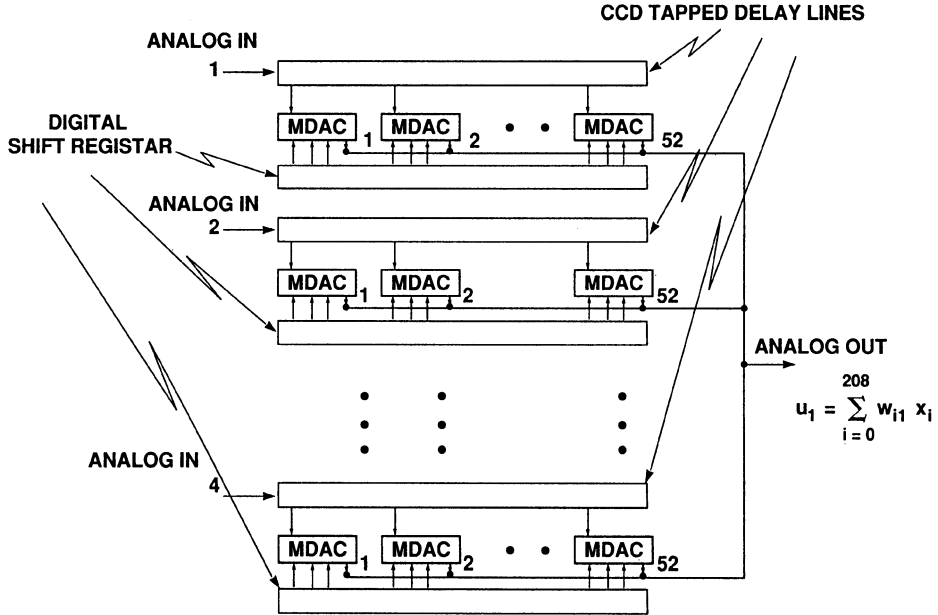


Fig. 4 CCD fully connected neural net processor with a single output node

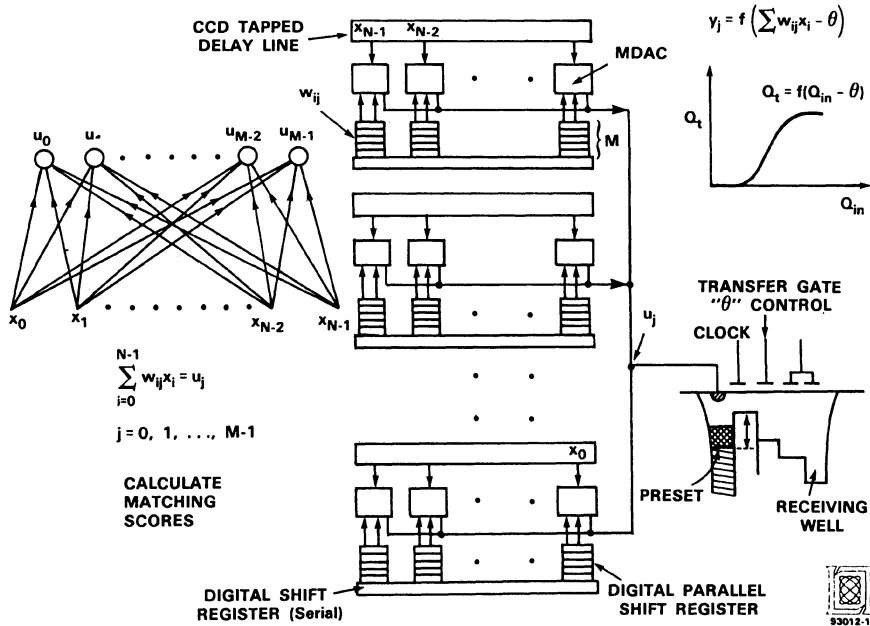


Fig. 5 CCD fully connected neural net processor with multiple output nodes.

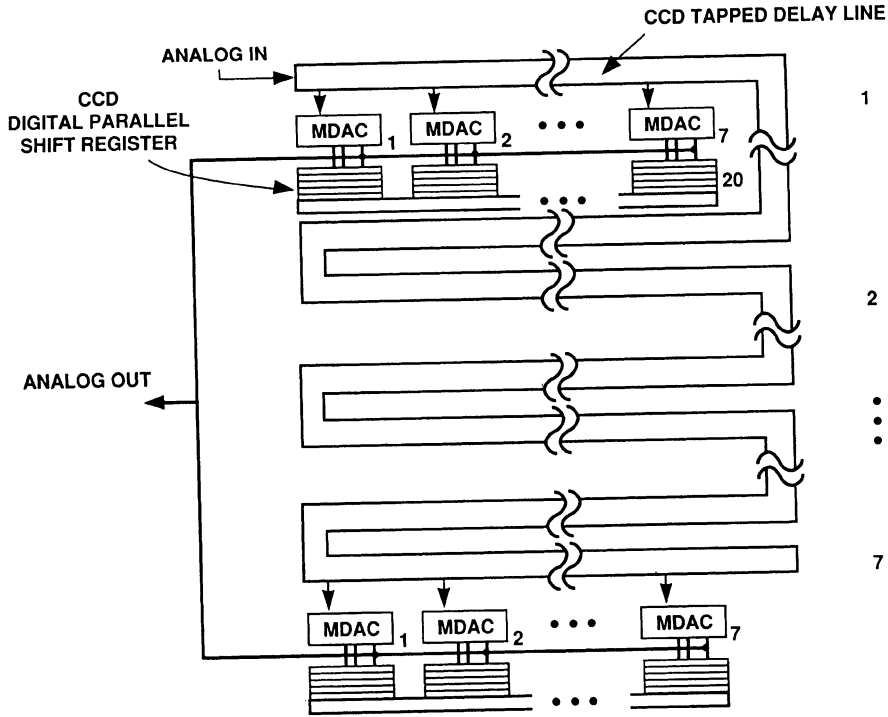


Fig. 6 CCD sparsely connected neural net processor.

$$Q_t = Q_{in} \left\{ 1 - \frac{0.97}{(\cos \tau_1)^{1/2}} \exp \left[-\frac{2.83}{\epsilon} \int_{-\tau_1}^{\pi/2} (\cos \tau)^{1/2} d\tau \right] \right\}$$

$$\epsilon = \left(\frac{CW}{AV_0} \right)^{1/2}, \quad -\frac{Q_{in}}{C} + V_0 \sin \tau_1 = V_T$$

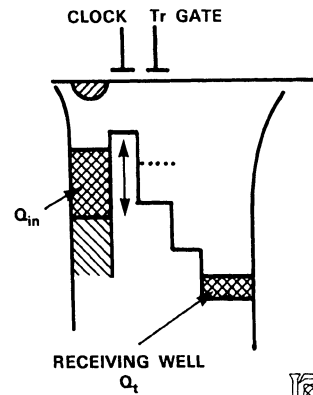
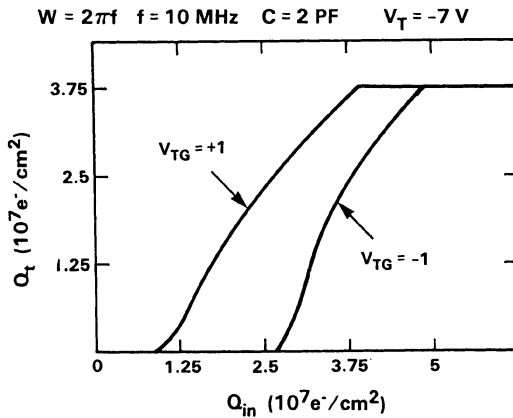


Fig. 7 Nonlinear dependence of the output charge on the input charge a bucket-bridge device for two different threshold levels.

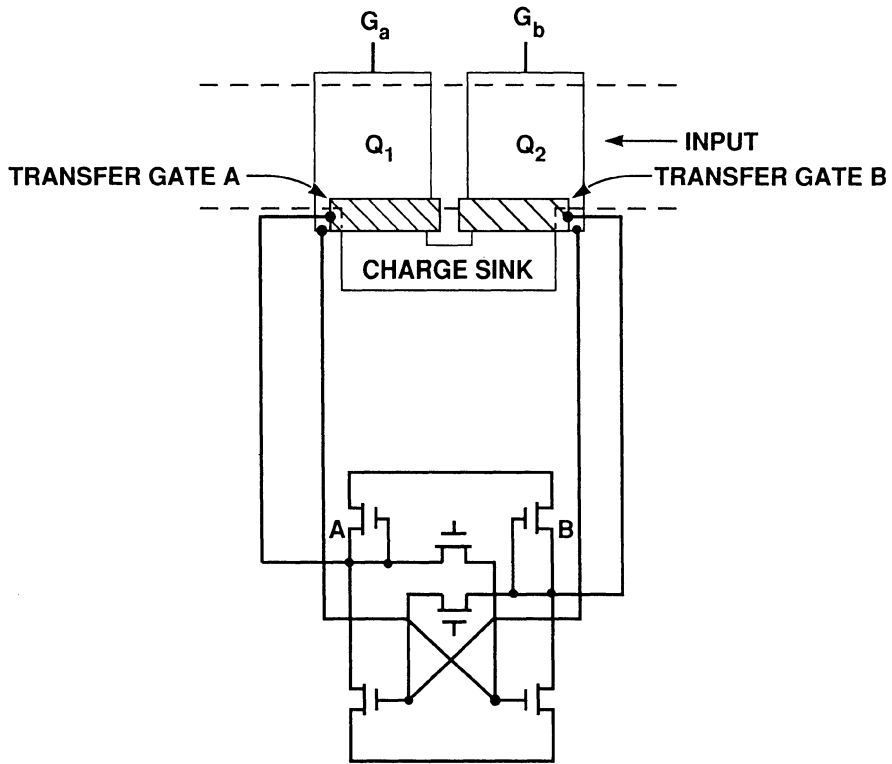


Fig. 8 Charge-domain maximum detection circuit.

Retina-Like CCD Sensor for Active Vision

G. Sandini¹, P. Dario², M. DeMicheli² and M. Tistarelli¹

¹ University of Genoa, Department of Communication, Computer and Systems Science - LIRA Lab,
via Opera Pia 11a, 16145 Genoa, Italy

² University of Pisa, Scuola Superiore S. Anna - ARTS Lab, and Centro "E. Piaggio", Pisa, Italy

Abstract. The amount of information contained in images is often redundant for the execution of specific tasks. In order to limit the information to a quantity compatible with the computational capabilities of feasible processing systems, it is essential to avoid or discard at early stages (possibly at the sensor level) excessive detail. Research in the field of active vision and sensory fusion are developing new theories exploiting the advantages of sensory motor coordination.

A fundamental problem for the reduction of visual data is to determine which part of image information is irrelevant before at least some pre-processing has been performed. In human vision, the structure of the visual sensor and the coordinated, task-driven, active control of eye movements have an essential role in the reduction of visual information. The distribution of the photosensitive elements in the human retina is such that the highest resolution is limited to the central part of the sensor (fovea) and decreases toward the periphery. The major advantage of this structure is the presence, on the same sensor, of a high-resolution sampling area and of a wide visual field. The exploitation of this structure in the context of different tasks is driven by the motor control strategies.

The design of a space-variant vision system and the relation to dynamic vision is discussed. A prototype space-variant CCD visual sensor is the base of the system, coupled with dedicated driving hardware and a motor sub-system. The CCD sensor is characterized by a resolution decreasing from the center toward the periphery which resembles the sampling structure of the human retina. Several vision modules are presented which take advantage of the space-variant structure of the sensor and the dynamic capabilities of the vision system. The implications derived from the use of this space-variant sensor in relation to visual processing are discussed.

1 Introduction

Visual sensors currently used in artificial vision are not optimally suited to many advanced robotics applications. The development of visual sensors, in fact, has been driven mostly by the "television" industry whose main concern, as far as TV cameras are concerned, is to acquire images with the highest possible accuracy. A great deal of technological research, for example, has been directed

toward the increase of spatial resolution to such an extent that currently adopted television standards are below technological limits, whereas, no research at all has been devoted to the problem of defining and developing visual sensors specifically for robotics applications.

From these considerations one could ask which are the distinctive features of a visual sensor in relation to robotic applications. The answer is not simple, given the fact that vision research itself is not so established to suggest global guidelines or specifications. However, considering the peculiarities of advanced robotics we could try to derive some general guidelines for visual processing. The starting assumption is that one relevant difference between artificial vision for robotics and, say, traditional image processing is the motion of the sensor: image processing is based on images acquired by static sensors or by sensors moving along known trajectories while in robotics (and particularly in advanced robotics) the sensor is moving and some components of its motion are unknown or too expensive to measure. Think, for example to a rover moving on rough terrain. In this case the motion of the cameras mounted on board depends very much on the profile of the terrain. Another example is a space manipulator performing assembly operation. The arm itself is floating and will need to be able to manipulate objects floating in space: it is not possible to "fix" the cameras or it may be too expensive. A possible solution to this problem is to design the robot systems in such a way to put the visual sensors in a controlled situation similar to that encountered in traditional robotic or image processing applications (like visual inspection). For example, to design active stabilization devices capable of maintaining a more or less constant orientation of the optical axis of the visual cameras. This could be the right approach if the stabilization technique is also based on visual information. Why this is important comes from two observations:

- the first one is that stabilization is mainly motivated by the need to acquire visual information "during motion" which is sufficiently "smooth" to obtain reliable velocity or velocity related measures.
- the second, and more important reason, is that vision could play an active role in the stabilization process.

Considering, for example, the structure of the human visual system it is clear that the stabilization of the optical axis of the eyes performed by the vestibulo-ocular reflex ¹ does not solve at all the problem of image stabilization unless a tracking process based on visual information is also activated. This, again, requires the selection of a world point to be tracked. ² In summary: what has been reported so far puts the accent on image tracking and visual stabilization as some of the basic distinctions between image processing and vision for advanced robotics. If this assumption is true our opinion is that currently used visual sensors are not well suited to such applications and, on the contrary, the structure of the sensor described in this paper can simplify this kind of processing very much.

¹the Vestibulo-ocular reflex is based on purely mechanical sensors (accelerometers) and its function is to maintain the optical axis of the eyes parallel to itself during motion of the head (caused either by direct motion of the head or by motion of the body)

²In fact, even considering a *simple* monocular system moving in a 3D environment, the displacements of the images of the objects depend upon their distance from the sensor. Therefore, it is impossible to track the entire image but a point need to be selected for tracking and only the points lying at the same distance of the fixated one will be stabilized (usually, the points around the fixation point). It is worth noting, however, that beyond a given distance from the sensor, the displacements of the image points do not depend very much on distance.

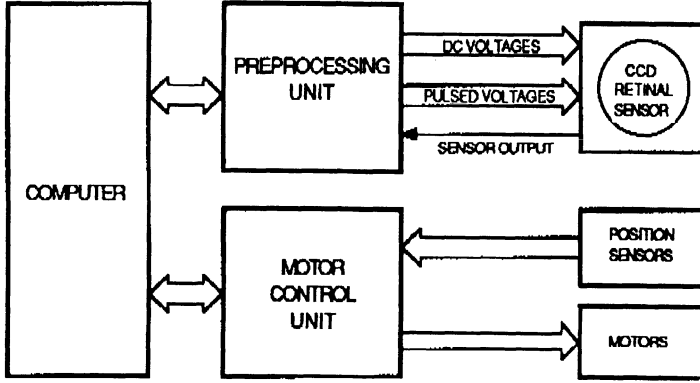


Fig. 1: Components of the Visual System

2 The image acquisition system

The developed vision system, sketched in Fig. 1, is composed of the following parts:

- a prototype camera with a retinal CCD sensor;
- a variable scan acquisition unit with driving electronics for the retinal chip;
- a motor sub-system to perform active movements controlled by visual feed-back.

All the system components are detailed in the following sections.

2.1 CCD-Retina and space-variant sensing

In the human visual system the receptors of the retina are distributed in space with increasing density toward the center of the visual field (the fovea) and decreasing density from the fovea toward the periphery. This topology can be simulated, as proposed by Sandini and Tagliasco [1], by means of a discrete distribution of elements whose sampling distance (the distance between neighboring sampling points) increases linearly with eccentricity from the fovea. An interesting feature of the space-variant sampling is the topological transformation of the *retinal image* into the *cortical projection*³ [1, 4].

This transformation is described as a conformal mapping of the points on the polar (retinal) plane (ρ, η) onto a Cartesian (cortical) plane $(\xi = \log \rho, \gamma = \eta)$, where the values of (ρ, η) can be obtained mapping the Cartesian coordinates (x, y) of each pixel into the corresponding polar coordinates. The resulting cortical projection, under certain conditions, is invariant to linear scalings and rotations on the retinal image.

Beyond the geometric properties of the polar and log-polar mapping, which will be referred to in the following sections, the log-polar transformation performs a relevant data reduction (because the image is not equally sampled throughout the field of view) while preserving a high resolution around the fovea; thus providing a good compromise between resolution and band-limiting needs.

³The terms "retinal" and "cortical" derive from the observation that the conformal mapping described here is very similar to the mapping of the retinal image onto the visual cortex of humans [2, 3].

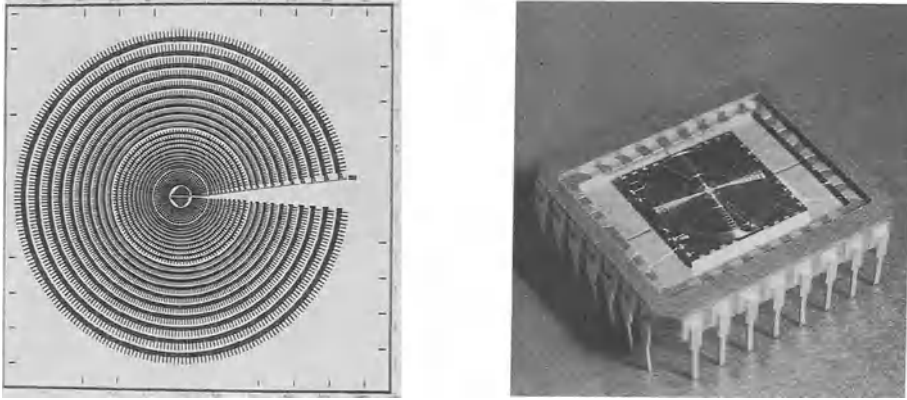


Fig. 2: (a) Structure of the prototype CCD retinal sensor. (b) Picture of the chip.

This property turns out to be very effective if you wish to focus attention on a particular object feature, or to track a moving target (i.e. to stabilize the image of the target in the fovea). Therefore the properties of the topological log-polar mapping has found interesting applications for object and, more specifically, shape recognition and object tracking [5, 6, 7, 8, 9, 10].

A prototype retina-like visual sensor has been designed within a collaborative project involving several partners ⁴. In this paper we will refer to the physical characteristics of the prototype sensor when dealing with the log-polar transformation. The results could be easily generalized to any particular log-polar mapping, by modifying the constant parameters involved in the transformation.

The retino-cortical mapping is implemented in a circular CCD array, using a polar scan of a space-variant sampling structure, characterized by the sampling period and the eccentricity (the distance from the center of the sensor). The spatial geometry of the receptors is obtained through a square tassellation and a sampling grid formed by concentric circles [11]. The prototype CCD sensor, depicted in Fig. 2, is divided into 3 concentric areas each consisting of 10 circular rows and a central fovea. Each circular row consists of 64 light sensitive elements [12]. The central fovea is covered by a square array of 102 light sensitive elements. ⁵ In the experiments the information coming from the fovea is not used.

As for the extra-foveal part of the sensor the retino-cortical transformation is defined by:

$$\begin{cases} \xi = \log_a \rho - p \\ \gamma = q \eta \end{cases} \quad (1)$$

where (ρ, η) are the polar coordinates of a point on the retinal plane, p , q and a are constants determined by the physical layout of the CCD sensor.

⁴The institutions involved in the design and fabrication of the retinal CCD sensor are: DIST - University of Genoa, Italy; University of Pennsylvania - Dept. of Electrical Engineering, (PA) - USA; Scuola Superiore "S. Anna" Pisa, Italy. The actual fabrication of the chip was done at IMEC, Leuven, Belgium.

⁵Currently the performances of the CCD sensor are being evaluated (the first "real" image has been recently acquired) and a prototype camera is being built. The experiments reported in this paper were carried out by re-sampling standard TV images following the geometry of the sampling structure of the sensor.

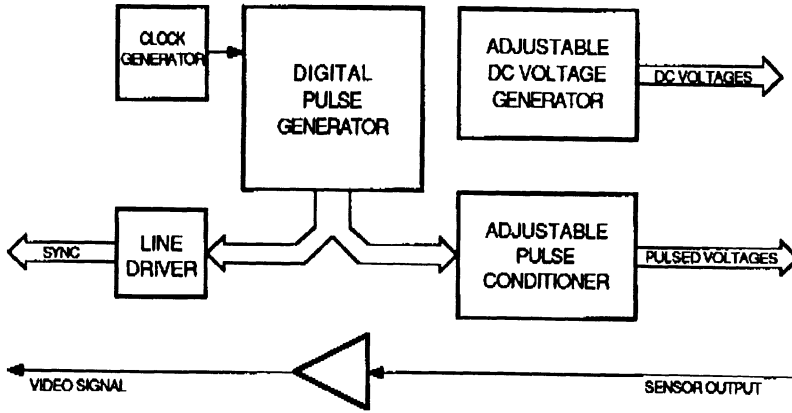


Fig. 3: Block diagram of the retinal sensor driving unit.

2.2 The pre-processing unit

In order to obtain the video signal from the sensor and to supply the appropriate sync signals for further digitalization, a dedicated unit was designed and developed.

The operation of the retinal sensor requires a set of pulsed voltages and a set of DC voltages. The pulsed voltages are needed to drive the circular CCDs, the coupler, the radial CCD and the output stage.

Since the circular CCDs are divided into three concentric areas, each working at a different frequency, and a CCD requires three clock phases to operate, 12 different clock voltages are needed. Three driving signals are necessary for the coupler, whereas one reset signal is required for the output stage. In addition, since the retina-like sensor includes a site for testing the CCD transport efficiency, one more driving voltage is necessary, yielding to a total of 18 pulsed voltages.

DC voltages are needed for appropriately biasing the output charge amplifier and the transfer gates. A total of 10 DC voltages are necessary.

The output signal is a sequence of square pulses, whose amplitude depends linearly on the amount of charge generated in the photosites.

The unit has been designed to be as flexible as possible, and to allow easy adjustments and tuning of the present prototype sensor. To this aim, the unit provides all the voltages (both pulsed and DC) the sensor requires, as well as the frame, radius, pixel and pixel-valid sync signals. A conditioning stage for the video signal is also provided. Furthermore, the unit can be driven by an external master clock. A block diagram of the unit is depicted in Fig. 3.

The unit comprises the following blocks, which correspond to the functions outlined above:

- a block (digital pulse generator) to generate the pulsed voltages in TTL format;
- a block to modify the shape parameters of the digital signals to meet the sensor's specifications (adjustable pulse conditioner);
- a block to generate the required DC voltages (adjustable DC voltage generator).

The block diagram also shows the conditioning stage for the sensor output. A master clock generator and a buffer stage for the sync signals are present.

Three 64Kx8 bit EPROMs are used in the digital pulse generator to generate all digital signals, including also sync signals. This provides flexibility to the unit and allows easy interfacing with an acquisition unit. In addition, individual/collective enabling for all generated signals, and direction reversing for the CCD clocks are provided.

The adjustable pulse conditioner allows trimming the high and low level of the digital signals, and setting the proper rise and fall times. In order to increase flexibility even more, all driving voltages are adjustable, whereas the clock phases of the same CCD have a collective trimming.

The video signal has a non-standard form. Thus, interfacing the unit (and hence the sensor) to a data-acquisition unit is not trivial. Signal acquisition requires a frame grabber capable of acquiring signals from non-standard imaging devices. The frame grabber must also have the ability to accept external sync signals. In addition, the EPROMs of the retinal sensor driving unit have to be properly programmed in order to generate the appropriate sync signals.

This has been done for a commercial variable-scan frame grabber. At present, an interface to a transputer-based frame grabber is under consideration. It will be possible to perform image processing without host intervention, while the processing power could grow as needed simply by adding more transputers using the available serial links. Furthermore, I/O TRAMs can be added to provide analog/digital input and output, giving the possibility to implement the whole motor control loop.

3 The motor sub-system

The concept of a retina-like, space-variant sensor is inherently related to the notion of active vision. In fact, the proposed CCD sensor would be useless without a mechanical system designed for controlling the motion of the sensor itself. In particular, a vision system based on the retinal sensor should be capable of controlling the orientation of the sensor visual axis in space, in order to fixate points, quickly move from one point to another (saccade), or track moving targets.

The human visual system is an obvious model for the motor subsystem we are considering. However, the performances of the human visual system are not trivial to replicate by a mechanical system, if size and weight are to be taken in consideration. Based on typical values known from the physiology of the human visual system [13], an artificial motor system should possess the following features: at least two rotational degrees of freedom per sensor; rotation: about 90 degrees per d.o.f.; peak angular velocity and acceleration: 500-600 *degree/s* and 13000-15000 *degree/s²*, respectively.

Rather than addressing immediately the problem of designing a mechanical motor system capable of the above performances, a relatively simple mechanical apparatus has been designed. A scheme of the apparatus is depicted in Fig. 4.

The apparatus consists of two specular parts, one for each sensor, each including two DC servomotors and two high resolution incremental encoders for position sensing. No reduction gears have been used in the servomotors, in order to avoid backlash and the consequent decrease of position accuracy of the CCD sensor. Appropriate reduction, with satisfactory accuracy at low cost, has been obtained by using no-slip pulleys and belts for transmitting motion from the motors to the frames which hold the CCD sensors.

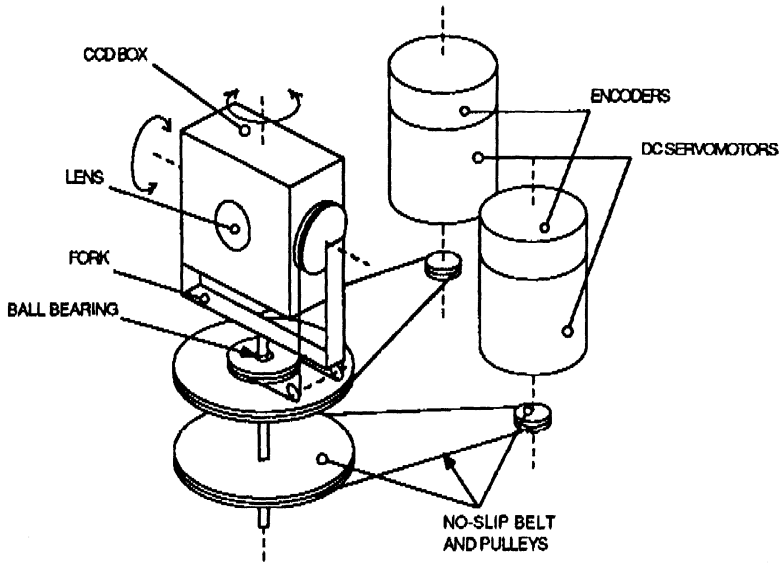


Fig. 4: Scheme of the motor subsystem.

An electronic unit has been designed for the low level control of the motors. No local loop exists between position sensors and motors: the overall control strategy will be implemented via software in order to achieve the flexibility required in the present experimental phase.

As far as the closed loop with visual processing is concerned, we are currently evaluating the performances of motion detection and target tracking algorithms, implemented on a 68020 based computer. The first task is to analyze a set of images and generate the control signals to smoothly track a moving target. Toward this aim we estimated to be necessary to process images and produce the control signals at a rate of about 10 Hz.

4 Visual processing

In order to verify and demonstrate the use of the space-variant sampling strategy in some active vision problems, device simulations have been performed. They include: active shape recognition, target tracking for active gaze control, vergence control, and estimation of the *time-to-impact*. This experimental activity has been possible thanks to simulation tools developed at DIST in the last years. The tools either run on a VDS 7001 *Eidobrain* image processing workstation or any Sun workstation. The timings recorded on these machines were indicative for a possible real-time hardware implementation of the algorithms.

4.1 Active shape recognition

A template-matching approach was used for the recognition of two-dimensional shapes independently from their position on the visual field, their spatial orientation, and their distance from

the sensing device. This procedure comprises two phases [5, 9]:

- a training phase, during which the objects belonging to the predefined visual world are sampled, moving the fovea over the center of mass of the object, and their cortical mappings are stored into a template file (because of the form-invariance property, only one template is stored for each object irrespective of size and orientation);
- and a recognition phase, where the retinal image is sampled, mapped into the cortical plane and segmented so as to separate the various objects; each object is then sampled again with the fovea over the center of mass of the object. The unknown object is then compared to the templates by computing a similarity score based on the cross correlation between object and templates.

The shape recognition algorithm has an obvious limitation in that the recognition is dependent upon the physical separation of the objects. The correct positioning of the fovea, before the template matching, is based on the computation of the center of mass. For overlapping 2D shapes the center of mass is no more an invariant feature. In this case, recognition is based on the result of successive template matching computed on different views, obtained by positioning the fovea over "peculiar" features of the objects. In our case, peculiar features are based on the corners along the boundary of the 2D object[9].

The exploration strategy, adopted to drive fixation, is of crucial importance [14]. In order to produce a set of useful characteristic views, gaze is sequentially moved over selected object features, in a manner similar to the behavior of the human oculo-motor system when trying to recognize an object [15].

4.2 Object tracking

The ability to identify and track a moving target is a fundamental aspect of active vision systems [16, 17]. Generally speaking, the tracking process can be decomposed in three phases:

- detection;
- foveation;
- and actual tracking.

Each phase has its own computational peculiarities related both to processing and to the constraints imposed on the system (refer to [6, 15] for a detailed description of the tracking algorithm).

The optical flow is computed to detect, isolate and foveate the target and then to track it. During this last phase, in order to estimate the tracking error, the expected optical flow, obtained from the known motion of the sensor, is subtracted from the optical flow computed from the sequence of cortical images. The tracking error is used to correct the rotational velocity of the sensor.

In order to test the performance of the algorithm, a synthetic image was generated by superimposing a textured pattern on a background from a real image. A sequence of images was obtained by moving the pattern on the background with a pre-defined motion law. The output of the target tracking is shown in Fig. 5(a) for a circular trajectory and Fig. 5(b) for a linear trajectory of the target.

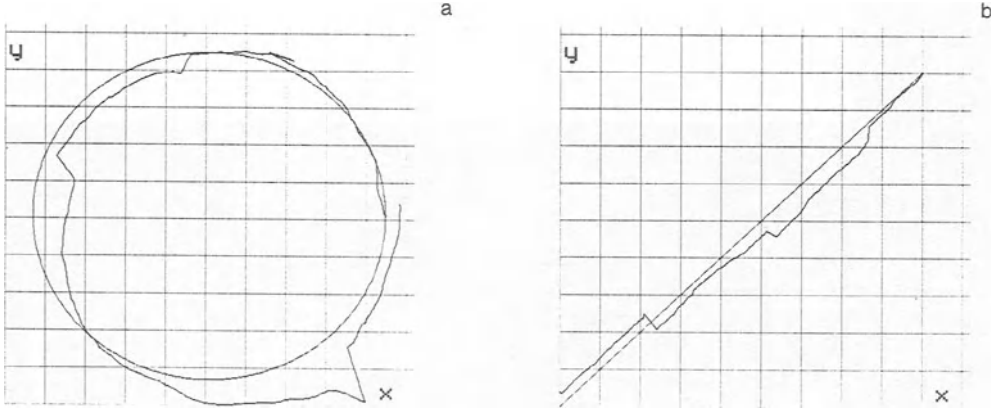


Fig. 5: Tracking of a textured target moving along a circular (a) and linear (b) trajectory. The trajectory of the fovea is superimposed to that of the target.

The optical flow is computed by solving an over-determined system of linear equations in the unknown terms $(u, v) = \vec{V}$. The equations impose the constancy of the image brightness over time and the stationarity of the image motion field [18]:

$$\frac{d}{dt}I = 0 \quad \frac{d}{dt}\nabla I = \vec{0} \quad (2)$$

where I represents the image intensity of the point (x, y) at time t . The least squares solution of (2) is computed for each point on the cortical plane.

4.3 Binocular vergence control

In binocular systems gaze and fixation control involve keeping the optical axes directed at the point in space currently fixated. This is accomplished through active vergence control.

Coombs and Brown [17] demonstrated a simple and efficient vergence control system based on cepstral filtering of stereo images. The rationale was the computation of a gross cross-correlation score between the left and right views, the maximum correlation identifying the correct vergence of the cameras.

A similar algorithm has been devised for space-variant stereo images sampled with the retinal sensor. The basic idea is that of computing a pointwise cross-correlation between cortical projections of the left and right views. The vergence between the cameras is varied by moving the *non-dominant* camera, while the cumulative cross-correlation is computed. When the cameras are almost correctly verged, the global cross-correlation becomes very high. In Fig. 6(a) a diagram is shown which illustrates the values of the inverse, global cross-correlation (simply computed as the normalized sum of the absolute difference of intensity values between the left and right image) obtained by applying the algorithm to the original, uniformly sampled images. As it can be noticed the outline of the diagrams in Fig. 6(a) and 6(b) is very similar, but the one obtained from the cortical images has a sharper peak at the correct vergence angle. This fact implies a faster convergence of the vergence control algorithm using images sampled with the retinal sensor. It is worth noting that the space variant sampling intrinsically emphasizes the relevance of the

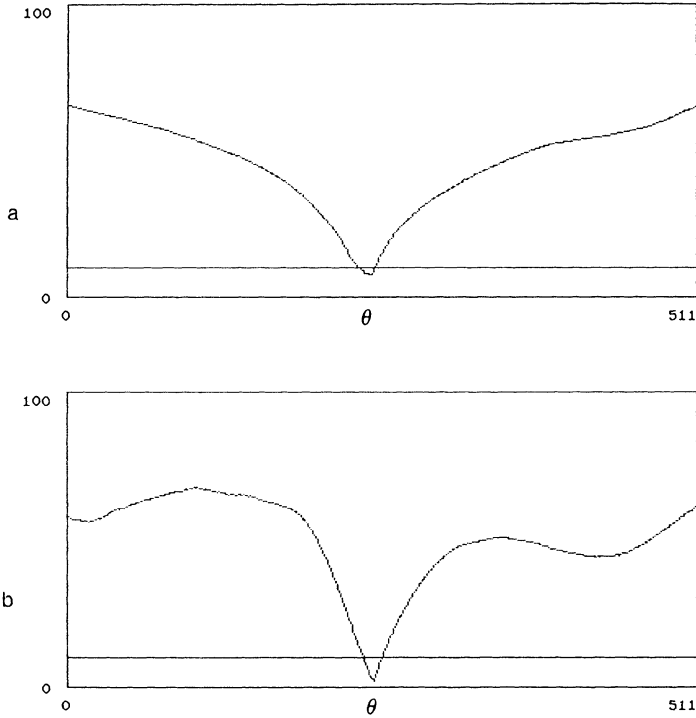


Fig. 6: Inverse, global cross-correlation of uniformly sampled (a) and space-variant (b) images, computed for different vergence angles θ .

central part of the image during the cross-correlation procedure (the number of pixels near or inside the fovea is much higher than the pixels on the periphery, therefore the central part of the image weights more than the periphery).

In Fig. 7 the images used to obtain the diagrams in Fig. 6 are shown. The image pairs corresponding to the maximum vergence and the correct vergence found using a gradient descent algorithm, together with the pixel-by-pixel inverse cross-correlation of the cortical images, is shown.

4.4 Motion and time-to-impact

The ability to quickly detect an obstacle to react in order to avoid it is of vital importance for animates. Passive vision techniques can be beneficially adopted if active movements are performed [16, 19, 20, 9]. A dynamic spatio-temporal representation of the scene, which is the *time-to-impact* with the objects, can be computed from the optical flow which is extracted from monocular image sequences acquired during *tracking* movements of the sensor [16, 21, 22, 23, 24].

Jain [25, 26] pointed out the advantages of processing the optical flow, due to camera translation, by using a log-polar complex mapping of the images and choosing the position of the FOE as the center for the representation.



Fig. 7: Results of the vergence control algorithm. On top are the original images, on the bottom are the cortical maps, with the result of the inverse cross-correlation in the middle. (a) Starting vergence. (b) Correct vergence found by the gradient descent algorithm. Notice that the values of the correlation are almost zero.

Generally, any expansion of the image of an object, due either to the motion of the camera or the object itself, will produce a radial component of velocity on the retinal plane. This intuitive observation can be stated in the following way :

the time-to-impact of a point on the retinal plane, only effects the radial component of the optical flow

This qualitative observation has been recently demonstrated [27]. It turns out that the most convenient way of representing and analyzing velocity is in terms of its radial and angular components with respect to the fovea [27].

Let us consider, for the moment, a general motion of the camera both rotational and translational. The image velocity on the (ξ, γ) plane, in case of small angular rotations can be expressed as:

$$\begin{cases} \dot{\xi} &= \frac{\dot{e}}{\rho} \log_a e \\ &= \left[\frac{1}{Z} \left[W_z - F \left(W_x \cos \frac{\gamma}{q} + W_y \sin \frac{\gamma}{q} \right) \right] + \left(\frac{\rho}{F} + \frac{F}{\rho} \right) \left(\phi \sin \frac{\gamma}{q} - \theta \cos \frac{\gamma}{q} \right) \right] \log_a e \\ \dot{\gamma} &= q\dot{\eta} \\ &= \frac{qF}{\rho} \left[\left(\frac{W_x}{Z} + \theta \right) \sin \frac{\gamma}{q} + \left(\phi - \frac{W_y}{Z} \right) \cos \frac{\gamma}{q} \right] + q\psi \end{cases} \quad (3)$$

These equations simply show that, while both components of the optical flow depend upon the depth Z of the objects in space, only the *radial* component $\dot{\xi}$ depends upon the time-to-impact $\frac{Z}{W_z}$. Moreover, only the *angular* component $\dot{\gamma}$ depends upon rotations around the optical axis, while the radial component is invariant with respect to ψ . Notice that up to now we have not made any hypothesis about the motion of the sensor and/or the objects in the scene. Therefore equations (3) certainly hold for any kind of camera and object motion.

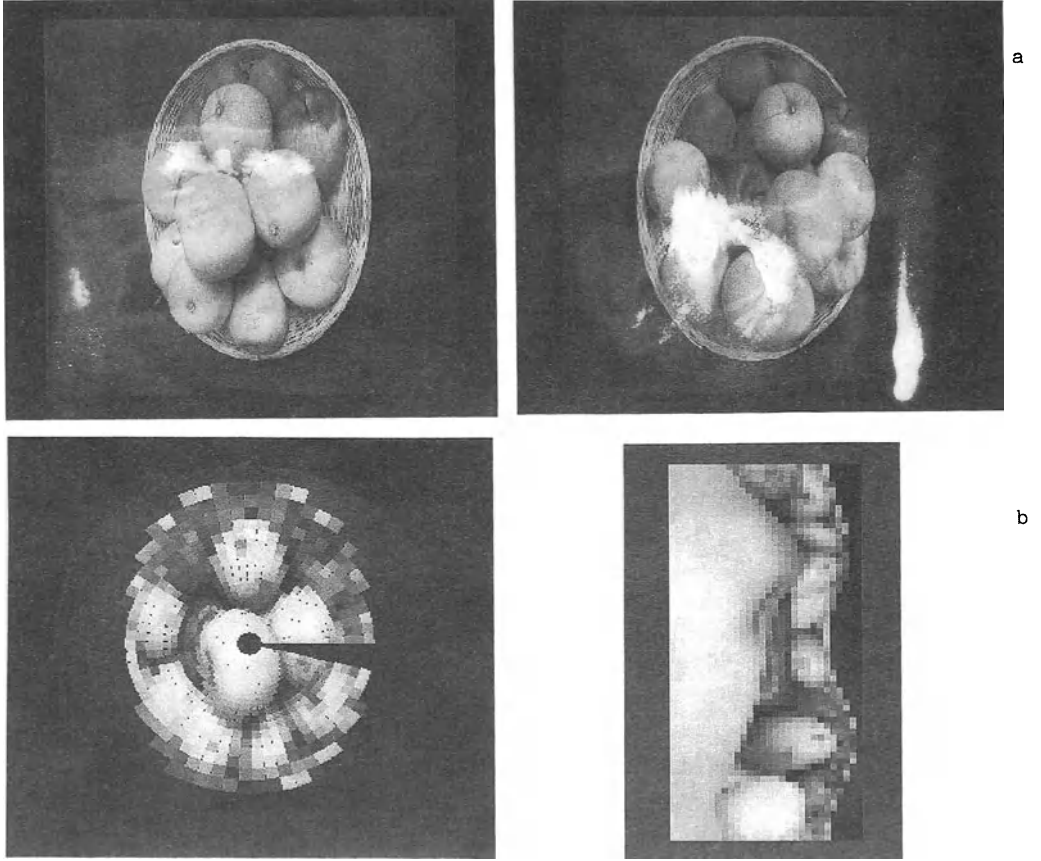


Fig. 8: (a) First and last image of the sequence. (b) Simulated output of the retinal CCD sensor for the central image of the sequence, represented in the Cartesian (x, y) and log-polar (ξ, γ) planes.

Equations (3) can be further developed in the case of tracking egomotion. By imposing $\vec{V}(0, 0) = \vec{0}$ in the general optical flow equations, we obtain:

$$\begin{cases} \dot{\xi} &= \left[\frac{w}{D} + \left[\frac{\rho}{F} + \frac{F}{\rho} \left(1 - \frac{D_2}{D} \right) \right] \left(\phi \sin \frac{\gamma}{q} - \theta \cos \frac{\gamma}{q} \right) \right] \log_a e \\ \dot{\gamma} &= \frac{qF}{\rho} \left(1 - \frac{D_2}{D} \right) \left(\theta \sin \frac{\gamma}{q} + \phi \cos \frac{\gamma}{q} \right) - q\psi \end{cases} \quad (4)$$

D_2 is the distance of the fixation point from the retinal plane measured at the frame time following the one where the optical flow is computed.

The time-to-impact of all image points can be easily computed from the partial derivatives of the optical flow [27]:

$$\frac{\mathbf{Z}}{\mathbf{W}_Z} = \left[\dot{\xi} \log_a e - \frac{\partial \dot{\xi}}{\partial \xi} + 2 \frac{\partial \dot{\gamma}}{\partial \gamma} \right]^{-1} \quad (5)$$



Fig. 9: (a) Optical flow of the sequence in Fig. 8(b), represented in the log-polar (ξ, γ) plane. (b) Time-to-impact of the scene in Fig. 8, computed by applying equation (5) to the optical flow in (b). For clarity, data is represented on the retinal plane.

Notice that only first order derivatives of the optical flow are required and the pixel position does not appear. The parameters q and a are calibrated constants of the CCD sensor. Equation (5) can be regarded as a formulation of the oriented divergence for the tracking motion, modified to take into account the fact that the sensor is planar and not spherical [28, 29].

In Fig. 8(a) the first and last image of a sequence of 10 is shown. The images have been acquired at the resolution of 256x256 pixels and then re-sampled performing the log-polar mapping. The motion of the camera was a translation plus a rotation θ around its vertical axis Y . The direction of gaze was controlled so as to keep the fixation on the apple in the center of the basket (which is the object nearest to the observer). The time-to-impact $\frac{W}{Z}$, computed by applying equation (5) to the optical flow in Fig. 9(a), is shown in Fig. 9(b). Data represented in Fig. 9 is relative to the 5th image of the sequence. Despite the low resolution the closest object can be correctly located. Note that most of the black patches correspond to areas where, due to the lack of structural information in the gray levels, image velocity has not been computed and consequently the time-to-impact has been set to zero ⁶.

Another equation for the time-to-impact can be obtained by computing the second order partial derivative of $\dot{\xi}$ [27]:

$$\frac{Z}{W_Z} = \left[\dot{\xi} \log_e a - \frac{\partial^2 \dot{\xi}}{\partial \xi^2} \log_a e \right]^{-1} \quad (6)$$

This equation clearly states that the time-to-impact can be computed using only the radial component of velocity.

⁶Note that a value of zero for the estimated time-to-impact is meaningless and it is interpreted as an unknown value of the real time until contact of the camera with the imaged object point. A real zero value could not be measured and, in any case, it would not be useful at all.

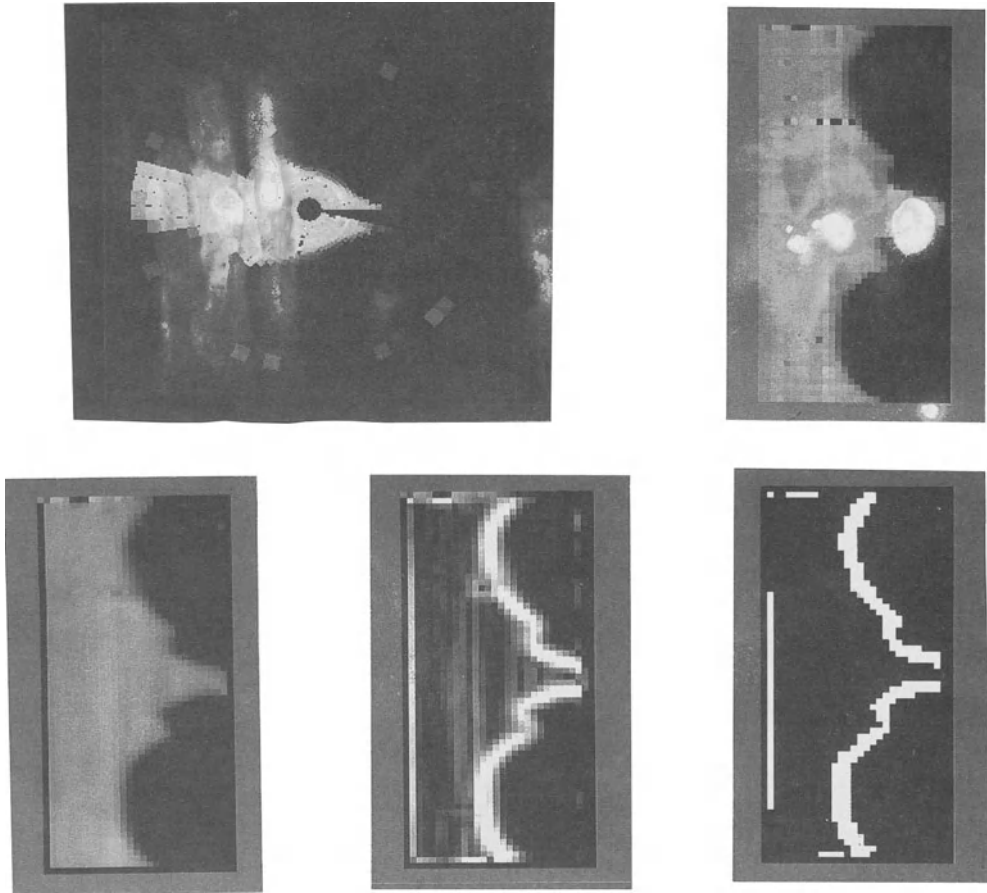


Fig. 10: Image of a binary shape (an arrow) acquired with the prototype CCD retinal sensor. a) Image on the retinal plane. b) Image on the cortical plane. c) Image after applying a 3x3 smoothing filter. d) Image after applying a Sobel filter for edge detection (amplitude of the intensity gradient). e) Edges obtained by thresholding (d).

5 Experimental results

Very recently the first images were acquired from a prototype camera with the CCD retinal sensor using the above described electronic unit and frame grabber. Following an experimental phase during which the characteristics of the sensor have been tested in terms of uniformity in sensitivity and linearity of response, very simple filtering algorithms have been implemented to start the real use of the sensor. In particular 3x3 histogram equalization and 3x3 filtering were implemented. In Fig. 10 and 11 the results obtained by applying a 3x3 smoothing filter and a Sobel edge detector to test images acquired with the prototype camera are shown. As it is clear, the overall performance of the chip is satisfactory and the sensitivity is rather homogeneous if we consider that the area of the outer photosites is about 14 times greater than those near the fovea. These images have been acquired with the very first version of the chip (Retina 1A) which, due to a

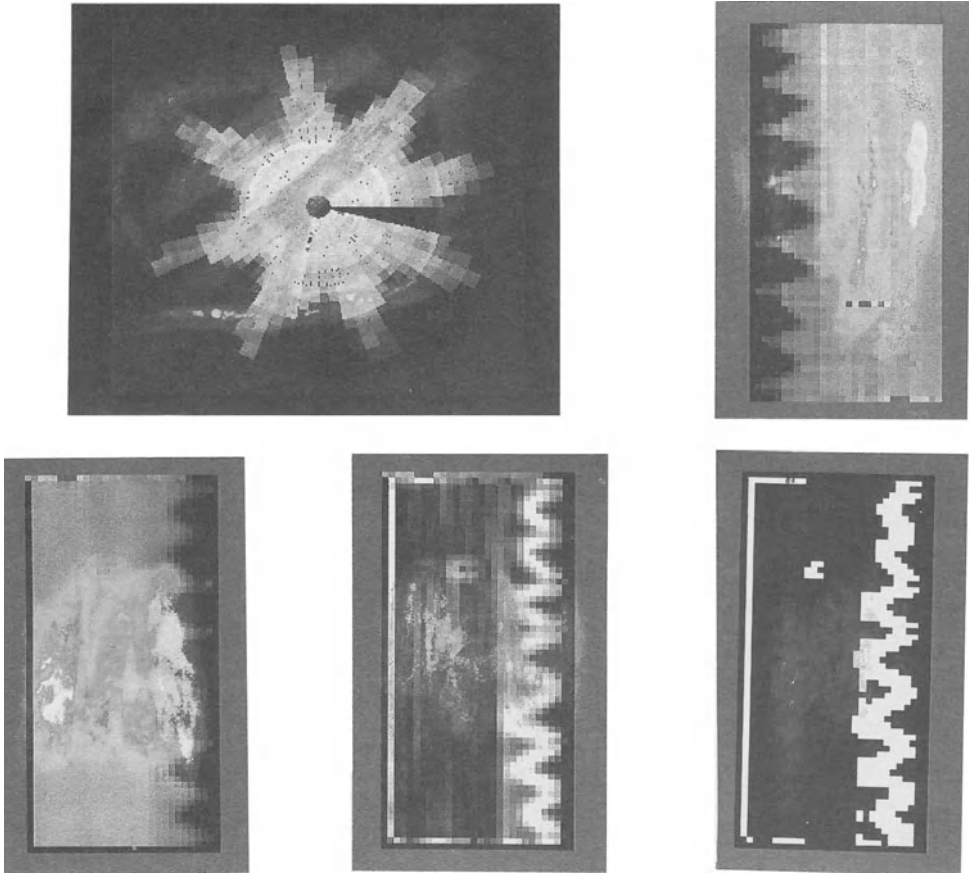


Fig. 11: Image of a binary shape (a star) acquired with the prototype CCD retinal sensor. a) Image on the retinal plane. b) Image on the cortical plane. c) Image after applying a 3x3 smoothing filter. d) Image after applying a Sobel filter for edge detection (amplitude of the intensity gradient). e) Edges obtained by thresholding (d).

problem in the mask making process, has one column (a circle) not properly working. Very recent experiments with the second version of the chip (Retina 1B) show, besides the correction of the problem, a more homogeneous sensitivity of the sensor.

6 Conclusion

It is our opinion that in the recent years much of the research on computer vision has been biased, as far as the sensor is concerned, by the requirements of "image processing" without stressing that in robotics images are to be understood usually from a moving platform (being it a head or a vehicle) and not transmitted or enhanced. This fact, which has been generally accepted as far as processing is concerned, has been mis-interpreted in relation to visual sensors.

Paradoxically, the latest developments in automatic video-cameras, have originated systems with

a built-in fovea. In fact, all the most advanced auto-focusing systems, ask the operator to point a small window, seen through the eyepiece, over the "important" feature he wants to focus on (this is equivalent to a fovea). It is important to note that it would be impossible to perform such operation, independently of the technology used to measure distance, without introducing this sort of foveal mechanism (i.e. it is impossible to focus "everywhere" at the same time). Vergence is another example (the two optical axes are to be oriented toward the same point in space). In this respect it may be said that *a fovea is necessary whenever a motor action is involved*.

A further point, which derives from the previous observations, is the misleading meaning which is sometimes given to the concept of *visual attention*. Usually, in fact, visual attention is related to the process of *interpretation* of the visual scene: "if I want to recognize a face I better shift my attention on the eyes and the mouth". Our point is that "attention" has also a very strong *behavioral* basis: "if I want to walk I better look in front of my feet". The major difference between these two different views is that while in the former case an a-priori hypothesis is necessary, based on some high-level processing or expectation, in the latter everything is based on self-generated actions which are, usually, task driven. In the examples given above both focusing and vergence involve the selection of a point in space *before the processing could even start*.

This behavioral connotation is stressed even more during tracking and for the computation of time-to-impact. In fact, if we consider the requirements of a tracking process, the only situation in which this process is not active is when the camera is fixed and looking to a steady environment. In all other situations the need to reduce motion blurring forces the activation of the tracking system which, consequently, cannot be actively suppressed. The alternative, proposed by the current technological trends, is to use high-speed shutters which "freeze" the image by sampling very shortly in time. This certainly avoids motion blurring but still does not prevent the overall system from measuring precise information about egomotion in order to be able to extract useful information from the evolving scene. Therefore, the peculiarities of a retina-like space-variant sensor in relation to visual tracking is, in our opinion, the major advantage of this approach: during tracking the high resolution part of the sensor is positioned over the object to track, i.e. where, in the image, the smallest displacements (actually the tracking errors) need to be measured while the low resolution part is sweeping over the background and motion blurring actually filters out the high frequency components of the incoming information.

A last observation is relative to depth and time-to-crash, crucial parameters in all robotic applications. To this respect the main point presented can be summarized by saying that, from a behavioral view point, depth computation is only necessary within the grasping range of the arm. As a consequence, during manipulative tasks the major role in depth computation is played by the stereo sub-system. Considering animals, for example, it is not a case that, the position of the eyes on the head, becomes frontal (i.e. maximizing the amplitude of the binocular visual field) as the manipulative ability increases.

Very different is the role of motion parallax for the control of ego-motion. In this case, in fact, the motion control system is not really interested in depth computation but in the computation of time-to-crash. In fact, in the case of a steady system (i.e. a system without legs or wheels) the computation of depth outside the range of grasping is entirely useless if one is interested in the role of vision for control and not for interpretation or recognition. On the other hand, as soon as the system starts moving the relevant information is not how far an object is but how long it will take until we reach it. The trajectory used to avoid an obstacle depends more on how fast we are approaching it than on its actual distance. For example, the steering strategy of a driver is very different when parking a car or passing a truck on a highway. Therefore, even if in principle

it is possible to compute depth from motion parallax, our opinion is that this is not necessary and that the simpler computation of time-to-crash is a lot more important in these cases. In the paragraph devoted to the use of the retina-like sensor for the computation of this behavioral variable we stressed the unique characteristics of this sampling strategy for the computation of time to crash.

7 References

- [1] G. Sandini and V. Tagliasco. An anthropomorphic retina-like structure for scene analysis. *CGIP*, 14 No.3 pages 365–372, 1980.
- [2] E. L. Schwartz. Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics*, 25 pages 181–194, 1977.
- [3] C. Braccini, G. Gambardella, G. Sandini, and V. Tagliasco. A model of the early stages of the human visual system: Functional and topological transformation performed in the peripheral visual field. *Biological Cybernetics*, 44 pages 47–58, 1982.
- [4] C.F.R. Weiman and G. Chaikin. Logarithmic spiral grids for image processing and display. *CGIP*, 11 pages 197–226, 1979.
- [5] L. Massone, G. Sandini, and V. Tagliasco. Form-invariant topological mapping strategy for 2-d shape recognition. *CVGIP*, 30 No.2 pages 169–188, 1985.
- [6] G. Sandini, F. Bosero, F. Bottino, and A. Ceccherini. The use of an anthropomorphic visual sensor for motion estimation and object tracking. In *Proc. of the OSA Topical Meeting on Image Understanding and Machine Vision*, 1989.
- [7] C. F. R. Weiman and R. D. Juday. Tracking algorithms using log-polar mapped image coordinates. In *SPIE Int. Conf. on Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, volume 1192, pages 843–853, Philadelphia (PA), 6-10 November 1990.
- [8] C. F. R. Weiman. Polar exponential sensor arrays unify iconic and hough space representation. In *SPIE Int. Conf. on Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, volume 1192, pages 832–842, 6-10 November 1990.
- [9] G. Sandini and P. Dario. Active vision based on space-variant sensing. In *Proc. 5th Int. Symposium on Robotics Research*, Tokyo, Japan, 1989. MIT-Press.
- [10] M. Tistarelli and G. Sandini. On the estimation of depth from motion using a ccd retinal sensor. In *Proc. of first European Conference on Computer Vision*, pages 211–225, Antibes (France), 1990. Springer Verlag.
- [11] J. Van der Spiegel, G. Kreider, C. Claeys, I. Debusschere, G. Sandini, P. Dario, F. Fantini, P. Bellutti, and G. Soncini. *Analog VLSI and Neural Network Implementations*. De Kluwer, 1989.

Acknowledgements: The authors thank F. Bosero, F. Bottino and A. Ceccherini for the help in developing the computer simulation environment for the CCD retinal sensor.

This research was supported by the Special Project on Robotics of the Italian National Council of Research.

- [12] I. Debusschere, E. Bronckaers, C. Claeys, G. Kreider, J. Van der Spiegel, P. Bellutti, G. Soncini, P. Dario, F. Fantini, and G. Sandini. A 2d retinal ccd sensor for fast 2d shape recognition and tracking. In *Proc. 5th Int. Solid-State Sensor and Transducers*, Montreux, 1989.
- [13] R.H.S. Carpenter. *Movements of the Eyes*. Pion Ltd., England, 1977.
- [14] H. Wechsler. *Computational Vision*. Academic Press, San Diego (CA), 1990.
- [15] G. Sandini and M. Tistarelli. Vision and space-variant sensing. In H. Wechsler, editor, *Neural Networks for Human and Machine Perception*. Academic Press, 1991 (in press).
- [16] D.H. Ballard, R.C. Nelson, and B. Yamauchi. Animate vision. *Optics News*, 15(5) pages 17–25, 1989.
- [17] D.J. Coombs and C.M. Brown. Intelligent gaze control in binocular vision. In *Proc. of 5th IEEE International Symposium on Intelligent Control*, Philadelphia, PA, September 1990.
- [18] S. Uras, F. Girosi, A. Verri, and V. Torre. Computational approach to motion perception. *Biological Cybernetics*, 60 pages 69–87, 1988.
- [19] R. K. Bajcsy. Active perception vs passive perception. In *Proc. Third IEEE Workshop on Computer Vision: Representation and Control*, pages 13–16, Bellaire (MI), 1985.
- [20] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *Intern. Journal of Computer Vision*, 1(4) pages 333–356, 1988.
- [21] G. Sandini and M. Tistarelli. Active tracking strategy for monocular depth inference over multiple frames. *IEEE Trans. on PAMI*, PAMI-12, No. 1 pages 13–27, 1990.
- [22] D. Raviv and M. Herman. Towards an understanding of camera fixation. In *Proc. of IEEE Int. Conf. on Robotics & Automation*, pages 28–33, Cincinnati, 1990.
- [23] M.A. Taalebinezhad. Direct recovery of motion and shape in the general case by fixation. In *Proc. of IEEE Intl. Conference on Computer Vision*, pages 451–455, Osaka (Japan), 1990.
- [24] D. Vernon and M. Tistarelli. Using camera motion to estimate range for robotic parts manipulation. *IEEE Trans. on Robotics & Automation*, RA-6, No. 5 pages 509–521, 1990.
- [25] R.C. Jain, S.L. Bartlett, and N. O'Brian. Motion stereo using ego-motion complex logarithmic mapping. *IEEE Trans. on PAMI*, PAMI-9, No. 3 pages 356–369, 1987.
- [26] S.L. Bartlett and R.C. Jain. Depth determination using complex logarithmic mapping. In *SPIE Int. Conf. on Intelligent Robots and Computer Vision IX: Neural, Biological and 3-D Methods*, volume 1382, Boston (MA), 4-9 November 1990.
- [27] M. Tistarelli and G. Sandini. On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. (submitted for publication), 1990.
- [28] W.B. Thompson and J.K. Kearney. Inexact vision. In *Proc. of IEEE Workshop on Motion: Representation and Analysis*, pages 15–21, Kiawah Island Resort, 1986.
- [29] R.C. Nelson and J. Aloimonos. Using flow field divergence for obstacle avoidance in visual navigation. *IEEE Trans. on PAMI*, PAMI-11, No. 10, 1989.

Designing Artificial Structures from Biological Models

Masuo Aizawa and Hiroaki Shinohara

Department of Bioengineering, Tokyo Institute of Technology
Nagatsuta, Midori-ku, Yokohama 227, Japan

Abstract

Several artificial structures have been designed on the model of the molecular communication mechanisms which are characteristic in the biological systems.

The first structure is an artificial synapse. The electrically stimulated release of a neurotransmitter from a presynaptic membrane has been simulated by the electric response of a polypyrrole membrane electrode. Anionic neurotransmitters such as glutamic acid are incorporated into the electrically oxidized polypyrrole film by electrochemical oxidation and released on call by electrochemical neutralization of the oxidized polypyrrole.

The second structure is a mechanochemical microgel with an electrical stimulator modeled beads are placed on the surface of the polypyrrole-coated electrode. Swelling and shrinking of these microgel beads are reversibly induced in accordance with controlled potential changes without any electrolysis of water.

The third structure is an optical biosensor for multimolecular information modeled on taste and olfactory mechanisms. The sensing element consists of an optically transparent substrate and several layers of LB films, each of which contains different fluorophors. The multi umami substances are successfully detected by response pattern in fluorescence in the whole wavelength range.

1. Introduction

Molecular communication is the characteristic information system in biological systems. The endocrine system may represent the features of molecular communication as illustrated schematically in Fig.1. The gland is a collection of specialized cells that synthesize, store, and release hormones. The hormone is released into the extracellular fluid and transported via the blood to two types of cells; target cells where the hormone acts, and other cells that degrade the hormone. In some systems, the target cell and the degradation site are in the same organ or are even the same cell. Both activities may even be located on the same plasma membrane. The receptor for the hormone is located on the surface of the plasma membrane.

Another example of molecular communication is found in a neuronal synapse, which is a communicating junction between two neurons. The presynaptic membrane releases the neurotransmitter that is bound by the receptor of the postsynaptic membrane.

The receptor molecular assembly may be classified into three categories based on its constituents:

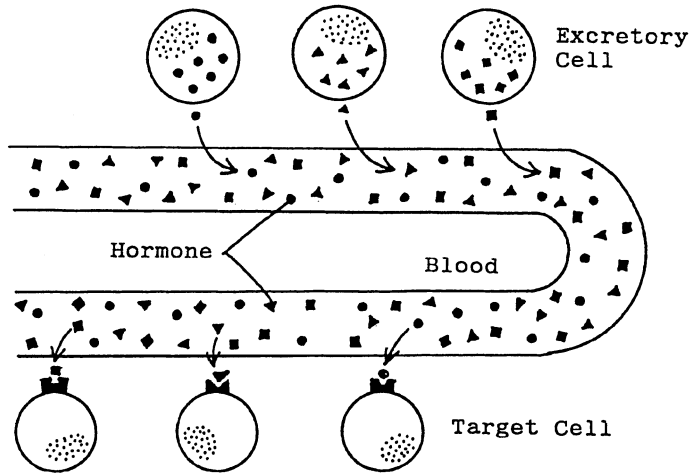


Fig.1 Schematic illustration of molecular communication in the endocrine system

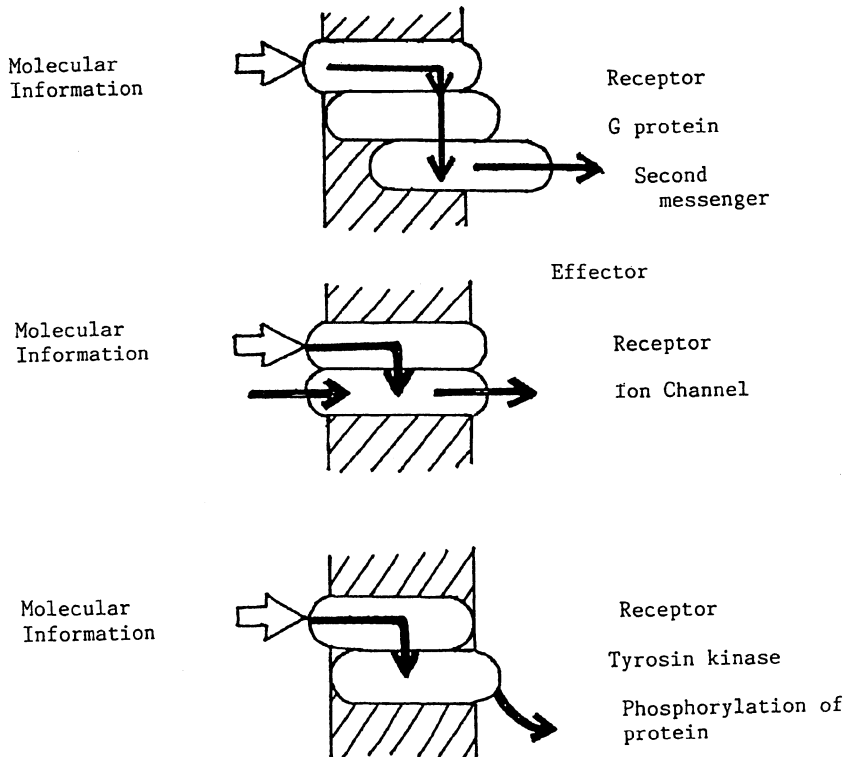


Fig.2 Schematic representation of molecular recognition by several receptors embedded in cellular membranes

1) Receptor → G protein → Effector (Adenylate cyclase etc.)

2) Receptor → Ion channel

3) Receptor → Tyrosin Kinase

The receptor protein recognizes and binds the corresponding molecules, such as neurotransmitters and hormones, which triggers activation of adenylate cyclase, ion channels, or tyrosin kinase as is schematically presented in Fig.2. The sensing function is systematically linked with the effector function in the molecular assembly. The effector function is completely modulated by the sensing function of the receptor protein.

This paper is concerned with designing artificial structures based primarily on molecular communication mechanisms seen in biological systems.

2. Electrically Stimulated Release of Neurotransmitter Modeled on Presynaptic Membranes

The electrically stimulated release of a neurotransmitter from a presynaptic membrane has been simulated by the electric response of a polypyrrole thin membrane electrode. Figure 3. shows the concept of electrically controlled release of anionic neurotrans-

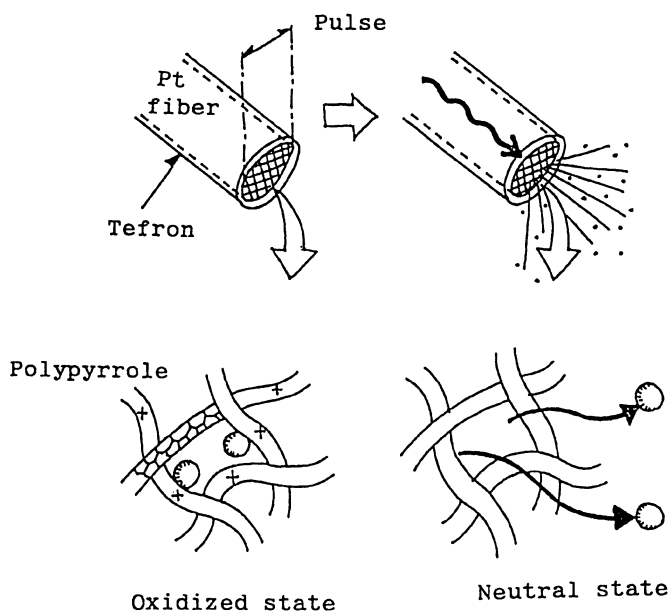


Fig.3 Conceptual scheme of electrically controlled release of neurotransmitters from a polypyrrole film based on the model of a presynaptic membrane.

mitters from a polypyrrole film based on the model of a presynaptic membrane. The presynaptic membrane releases a neurotransmitter in response to an electrical impulse. Based on the model of a presynaptic membrane, anionic neurotransmitters such as glutamic acid are incorporated into the electrochemically oxidized polypyrrole film by electrochemical neutralization.

Electrochemically stimulated release of anionic neurotransmitter was demonstrated with a microtip of polypyrrole film deposited on a platinum fiber [1-3].

A polypyrrole thin film was prepared on a platinum wire ($D=50\mu\text{m}$, $L=5\text{cm}$) by the electrochemical oxidation of pyrrole. The electrolyte solution contained 0.1 M pyrrole and 0.1 M tetrabutylammonium perchlorate in acetonitrile. Polymerization was carried out at a controlled potential of 1.0 V vs. Ag/AgCl. A polypyrrole film was grown up to about 200nm in thickness by controlling passed charge. The polypyrrole-coated platinum electrode was thoroughly washed with acetonitrile and distilled water, and then immersed in a borate buffer solution (pH 9.0) containing anionic neurotransmitters such as glutamic acid, aspartic acid, γ -aminobutyric acid and glycine. In this experiment, a pH 9.0 borate buffer solution was used as a standard solution in all measurements, because the above transmitters might dissociate to give negative charges. Electrochemical incorporation and release of anionic neurotransmitters were carried out by potential control with a potentiostat and a function generator (EG & G Princeton Applied Research, Model 363 and 175).

Cyclic voltammetry and differential pulse voltammetry showed that anionic neurotransmitters such as glutamic acid could be incorporated and released from a polypyrrole micro-tip, when it was soaked in a pH 9.0 borate buffer solution containing 10mM glutamic acid. Anodic current was derived from the oxidation of polypyrrole which resulted in the incorporation of anionic neurotransmitters into the polypyrrole matrix. On the other hand, cathodic current arose from the reduction of the oxidized polypyrrole that caved the incorporated neurotransmitter to be released. The incorporation and release of neurotransmitters was reversibly repeated by the electrochemical redox reaction.

The electrochemical release of neurotransmitters was confirmed by the method of fluorescence labeling. The polypyrrole-coated electrode was immersed in a solution containing glutamic acid and borate buffer. The electrode potential was set at 0.15V vs. Ag/AgCl so as to incorporate glutamic acid into the polypyrrole matrix. The electrode was thoroughly rinsed with borate buffer and soaked in the same buffer. The electrode potential was shifted to -0.4V vs. Ag/AgCl in order to release the incorporated glutamic acid from the polypyrrole matrix. The buffer solution was assayed for glutamic acid by the fluorescence method using fluorescamine before and after the electrochemical release. Fluorescamine reacts with primary amines to produce fluorescent products. The assay indicated that glutamic acid was electrochemically released from the polypyrrole matrix.

Timed release of glutamic acid was then clearly demonstrated with the polypyrrole microtip device that was prepared using the model of a presynaptic membrane. The microtip, which had electrochemically incorporated glutamic acid, was subjected to a -350 mV

voltage pulse with a 1 millisecond (ms) width. Glutamic acid was released on call from the polypyrrole microtip in response to the electrical stimulation of a 1ms pulse.

The polypyrrole microtip device modeled on a presynaptic membrane may be the first step toward realizing an artificial synapse or neuron. Although the present device is a prototype, it can release neurotransmitters on call, a characteristic which will find various applications.

3. Mechanochemical Changes Induced by Electrically Controlled Molecular Stimulation

Molecular communication is also found in synapses at a neuromuscular junction. An impulse conducted via neurons reaches the synapse and results in the release of molecular messengers such as Ca^{2+} , which trigger the contraction of muscle fibers. The mechanism of muscle fiber contraction influenced us to design a soft actuator controlled by molecular stimulation. In our investigation, this new system was achieved by combining mechanochemical microgel beads and an electrical stimulator which undergoes controlled release of molecular messengers [4-7].

Since a polypyrrole film-coated electrode showed excellent performance regarding the electrically controlled release of neurotransmitters, the electric stimulator was assembled by coating a platinum plate with a thin membrane of polypyrrole. A polypyrrole thin film was deposited on the platinum electrode surface by the electro-oxidative polymerization of pyrrole in an aqueous solution at a constant potential of 0.8 V vs. Ag/AgCl. The electrolyte solution contained 0.1 M pyrrole and 0.1 M NaCl. The film was grown up to about 200 nm in thickness.

The pH on the surface of the polypyrrole was directly measured with a micro pH probe. The potential of the polypyrrole-coated electrode was first controlled at a potential of -0.6 V vs. Ag/AgCl to neutralize the polypyrrole and undope anions. The pH at -0.6 V was 10.3. When the potential was shifted to 0.6 V vs. Ag/AgCl, the pH immediately changed to 3.7 within 90 sec. On the other hand, no appreciable pH change was observed on the surface of a bare platinum electrode under the same potential change. These results indicated that dynamic ion flux could be induced by a small change of potential. It is also emphasized that the ion flux occurred only in the vicinity of the surface.

Mechanochemical microgel beads were placed on the surface of the polypyrrole-coated electrode and the electrode potential was controlled with a potentiostat. Swelling and shrinking of these microgel beads was repeatedly induced on the surface of the polypyrrole-coated electrode in accordance with controlled potential changes. The potential change causes the redox state of the conductive polymer layer to shift in pH. Such a shift in pH could result in a drastic mechanochemical change of these microgel beads as is schematically illustrated in Fig. 4.

Electrically induced mechanochemical changes of polymer gels have been performed by many researchers. However, the electric mechanochemical changes are commonly associated with electrolysis of water at the electrodes, because it is needed to apply higher voltage. On the other hand, the mechanochemical changes may be

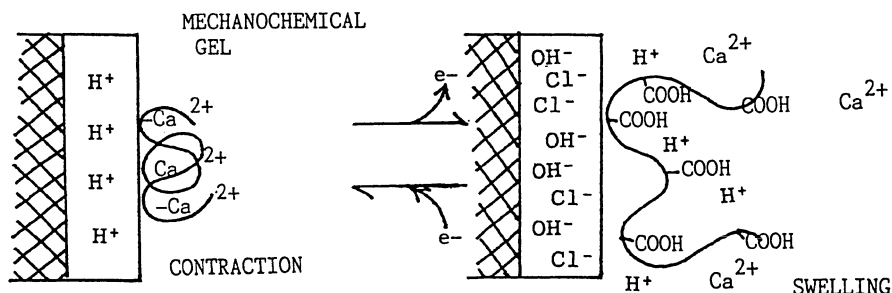


Fig.4 Electrically induced mechanochemical change of microgel beads

induced electrically without any electrolysis of water, if the polymer gels are electrically stimulated with the conductive polymer-coated electrode.

4. An Optical Biosensor for Multi-molecular Information Modeled on Taste and Olfactory Sensory Mechanisms

Taste and olfactory cells simultaneously accept multimolecular information, which is followed by recognition of smell and taste through sophisticated information processing in the nervous system. Signals generated from taste and olfactory cells do not seem to be specific for the corresponding substances. These signals, however, are simultaneously transferred through neuro-networks and finally excite neurons in the brain where taste and olfaction are recognized by the pattern of excited neuron distribution (Fig. 5).

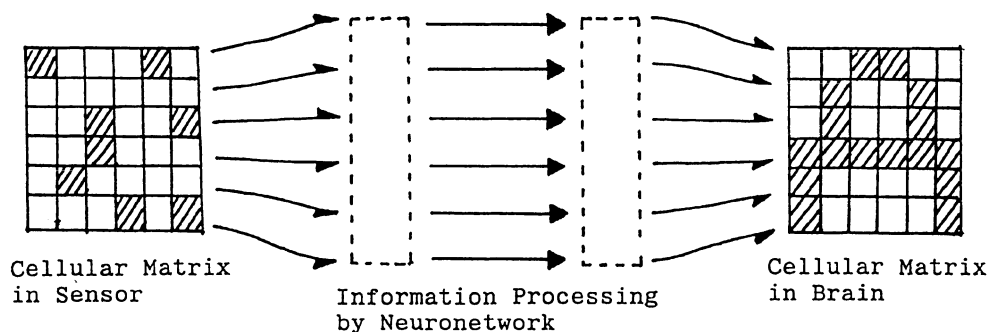


Fig.5 Pattern recognition in taste and olfactory information processing

There are two approaches in realizing a taste sensor. One approach is to integrate many selective sensor elements for multimolecular information. The other approach is to integrate non-selective sensor elements. Both types of sensors require information processing for the simultaneous recognition of multiple components. In this investigation, non-selective optical sensor elements have integrated to simulate the function of the taste sensory system.

The concept of an optical biosensor for multi-molecular information is schematically illustrated in Fig. 6,[8,9]. The sensing element consists of an optically transparent substrate and several layers of Langmuir-Brodgett (LB) films, each of which contains different fluorophors. Fluorescence of a LB film may be quenched or enhanced when the film comes in contact with the corresponding quencher or enhancer. Since each fluorophore fluo-

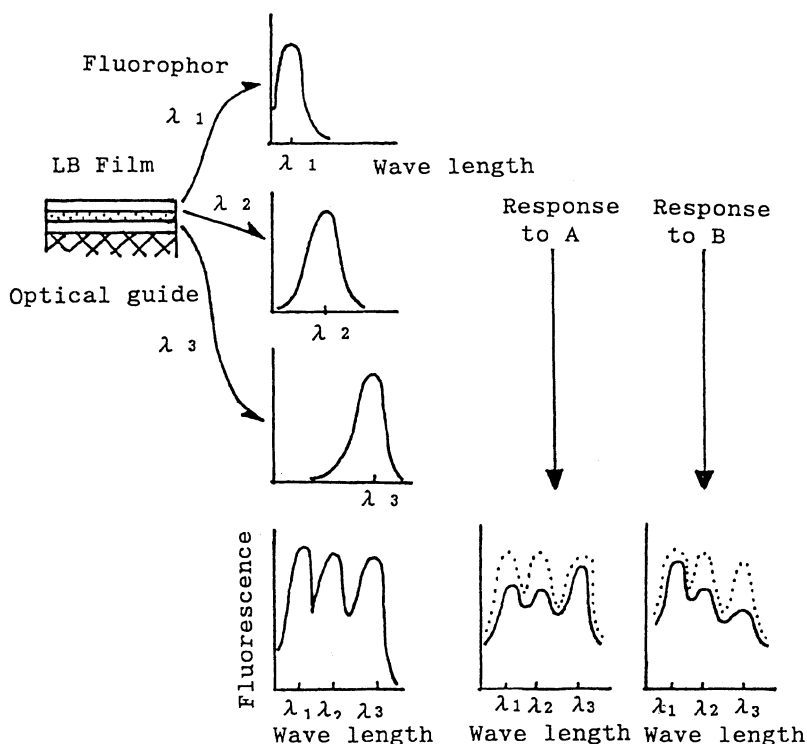


Fig.6 Concept of an optical biosensor for multi-molecular information

resces at a different characteristic wavelength, the response of the film can be differentiated.

A solution containing 9,12-anthroyloxy stearate and stearate (molar ratio 0.2) was spread on the water surface of a 0.25 mM CdCl_2 aqueous solution in a trough (Kyowa Kaimen Kagaku Co. Ltd.). Two layers of anthroyloxy stearate and stearate composite films were deposited on a non-fluorescent quartz glass plate at a surface pressure of 27.5 dyn cm^{-1} . Two layers of stearate films were then deposited. In addition, four layers of perylene/arachic acid films, four layers of butyrate/arachic acid (molar ratio 0.20), and two layers of arachic acid films were sequentially deposited at a surface pressure of 25.0 dyn cm^{-1} . Each fluorophore embedded in a different layer showed a characteristic fluorescence peak.

The sensor was placed in contact with a solution containing umami substances such as ATP, AMP, GMP, IMP and MSG. These umami substances gave intensive quenching effects on the fluorescence of the LB films. The characteristic fluorescence quenching effects were induced by each umami substance. Therefore, umami substances can be recognized by the characteristic pattern of fluorescence response of the sensor.

Umami is extremely enhanced when nucleotides such as GMP and IMP are in the presence of MSG. Such a multiple effect is an important factor in tasting delicious foods. A very unique characteristic was derived in response to the coexistence of GMP and MSG. The two umami substances enhanced the fluorescence in the wavelength range of 360-410 nm, although each of them gave a quenching effect if it is present alone. The multiple umami effect was successfully detected on the base of this fluorescence enhancement. The components of the umami substances were identified by the response pattern in other wavelength regions.

References

- 1) H. Shinohara, M. Aizawa, H. Shirakawa, Chem. Lett. 1985 179 (1985)
- 2) H. Shinohara, A. Aizawa, H. Shinohara, J. Chem. Soc. Japan 1986 465 (1986)
- 3) H. Shinohara, M. Aizawa, H. Shirakawa, J. Chem. Soc. Chem. Commun. 1986 87 (1986)
- 4) M. Yaoita, H. Shinohara, M. Aizawa, Y. Ikariyama, Bioelectrochem. Bioeng. 20 169 (1988)
- 5) H. Shinohara, J. Kojima, M. Yaoita, M. Aizawa, Bioelectrochem. Bioeng. 22 23 (1989)
- 6) H. Shinohara, J. Kojima, M. Aizawa, J. Electroanal. Chem. 266 297 (1989)
- 7) H. Shinohara, M. Aizawa, J. Polymer Soc. Japan 46 703 (1989)
- 8) M. Aizawa, M. Matsuzawa, H. Shirakawa, Thin Solid Films 160 477 (1988)
- 9) M. Aizawa, K. Owaku, M. Matsuzawa, H. Shinohara, Thin Solid Films 161 179 (1989)

DESIGN STRATEGIES FOR GAS AND ODOUR SENSORS WHICH MIMIC THE OLFACTORY SYSTEM

Krishna C. Persaud, Jon Bartlett and Paolo Pelosi*

Department of Instrumentation and Analytical Science,
UMIST, Manchester M60 1QD, UK

1. INTRODUCTION

The methods of detection, identification and measurement of concentrations of gases and odours have been under constant development for the last three decades. However, the numerous patents for new gas sensors appearing every year indicate that the state of the art is still not very satisfactory. For detection of individual gases and odours of industrial interest the following methods are widely used.

- (i) Use of specific chemical reactions coupled to visual indicator dyes. (For single use only). (Draeger Ltd, UK).
- (ii) Bulk effect metal oxide based gas sensors for general purpose detection of reducing gases. (Relatively non-specific, poisoned by some gases, multiple use). (Taguchi Ltd, Japan).
- (iii) Platinum pellistor elements for detection of combustible gases. (Non-specific, easily poisoned, multiple use).
- (iv) Polymeric silicon gas sensors with each individual sensor doped to detect a particular gas with high specificity. (Reversible, specific, multiple use). (ECO-SEMA Ltd, UK)
- (v) Gas-liquid chromatography coupled to one of the above detectors or to mass spectrometry. (Highly specific and accurate).
- (vi) Panels of trained humans for evaluation of quality and intensity of odours. (Highly subjective, but capable of evaluating complex mixtures).

Each of these gas and odour detection systems can be used effectively for particular applications. However, as detectors of volatile chemicals, they fall far short of the human olfactory system in terms of sensitivity, capability of discrimination of complex mixtures and flexibility in adapting to ever changing environments. A device capable of mimicking the olfactory discrimination mechanisms would be useful to many industries, not only for detection of particular gases of interest, but for evaluation of mixtures, such as for quality control in the food and beverage industry, or environmental monitoring.

* Istituto di Industrie Agrarie, Universita degli Studi, Pisa, ITALY

The biochemical mechanisms involved in olfactory perception and odour discrimination are not yet fully understood. However, the rapid progress of olfactory research in recent years now allows feasible design strategies for electronic gas and odour detection systems. It is not necessary or even possible to copy the biological system exactly. However, the biological design specifications of odour transducers, and some of the information processing concepts built into the olfactory system can be incorporated into workable electronic designs. Several researchers have been active in this area over the last twenty years. They include Dravnieks (1965), Tanyolac (1968) Herberhold (1969), Persaud and Dodd (1982), Abe *et al.* (1988) who have all attempted to produce devices which incorporated some of the concepts of the biological olfactory system.

This article presents our recent work in the design of gas and odour transducer elements and algorithms used for information processing, using loosely borrowed concepts from the olfactory system. We have chosen to depart from the established methods because of their technical limitations and to develop new types of gas and odour transducers. Some organic conducting polymers have unique electrical characteristics and show a reversible change in conductivity in response to certain gases. Unlike most other gas sensors which work at high temperatures and are easily poisoned by common gases, the new materials hold a promise of highly sensitive gas sensor elements with characteristics resembling the specifications of olfactory receptors. The relevant properties of the biological chemoreception system are outlined, feasible artificial analogs of receptors and information processing algorithms are given and a prototype device incorporating some of the discrimination mechanisms of the biological system is presented.

2. THE SENSE OF SMELL IN VERTEBRATES

The olfactory system may be divided into three parts (Fig.1):

(a) The Olfactory Epithelium

The olfactory epithelium presents an exposed odour-sensitive surface to the external environment. This contains the receptor system. In man, the olfactory epithelium covers about 6 cm². in each nostril. The surface contains about 5×10^7 receptor neurons, embedded in supporting structures. These cells contain cilia which project into the mucus overlying the epithelium, thus greatly increasing the surface area of the cells. There is evidence that these structures contain the olfactory receptor proteins. The serous glands, which provide the liquid medium overlying the olfactory epithelium, are located in the submucosa.

The chemical stimulus alters the properties of a specialized area of the receptor cell membrane, such that the ionic permeability changes and a change in membrane conductance occurs.

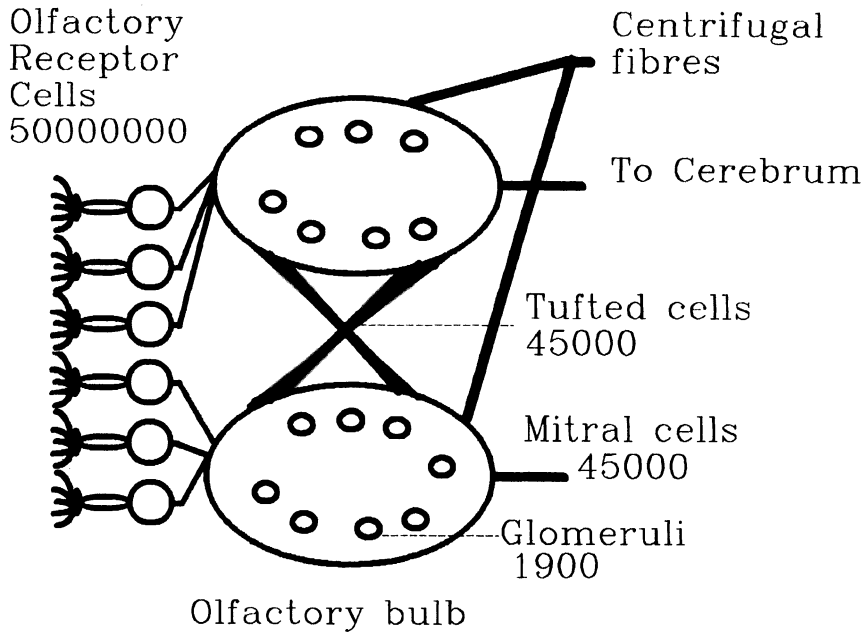


Figure 1 Schematic diagram of the olfactory system.

This features a highly convergent pathway. About $5 \cdot 10^7$ olfactory neurons send signals to about 1,900 glomeruli in the olfactory bulb, where substantial pattern deconvolution takes place. The signals are carried to the central nervous system via mitral cells and about 45,000 neural connections. Cross connections to the other olfactory bulb also exist via tufted cells. Substantial internal feedback systems exist, and many fibres from various parts of the central nervous system modulate the activity of the olfactory bulb.

The change in conductance, together with the electrochemical gradient across the membrane, cause an inward current (receptor or transduction current). The summated change in current across the olfactory mucosa shows an approximately logarithmic relationship to the concentration of the stimulus, the range of identifiable concentrations extending over 2-3 log units.

The receptor potential produces a change in membrane potential (receptor potential). This spreads to an area of membrane that has a regenerative current-voltage relation. This is the spike-initiating region located on the distal part of the olfactory neuron. The electronic spread of the receptor potential is called the generator potential.

(b) The Olfactory Bulb

The olfactory mucosa is separated from the brain by a bony structure called the cribriform plate. This contains holes which allow olfactory nerve fibres to pass through. Just above the cribriform plate is an extension of the anterior portion of the brain which is the olfactory bulb.

The neural spikes generated in response to an odour stimulus are transmitted to the olfactory bulb via the synapses of the olfactory nerve. The olfactory bulb is an important component in the processing of signals from the olfactory epithelium. The primary olfactory neurons pass in bundles to the olfactory bulb without intervening synaptic connections. A distinct anatomical feature of the olfactory bulb is a high degree of neural convergence. The 5×10^7 olfactory neurons connect to about 45,000 mitral cell axons of the olfactory tract. The primary neurons converge in groups of about 25,000 upon about 1,900 relatively large structures called glomeruli. The neural impulses received by each glomerulus pass through about 24 mitral cells and about 68 tufted cells. The latter provide a link between the two olfactory bulbs, so that the epithelium from one nostril has anatomical connections with both halves of the brain. Feedback connections also exist from the higher processing centres.

(c) The Cerebral Hemispheres

In the cerebral hemispheres, the inputs from the olfactory bulb are combined with inputs from the other senses and with feedback systems within the brain. At this level, the message that is generated by the olfactory epithelium in response to an odour, is recognised as such, and appropriate behavioural action is initiated if necessary.

The olfactory bulb output is carried in the mitral and tufted cell axons and distributed to several basal forebrain regions. Apart from those pathways directly involved in sensory processing of olfactory input, each olfactory region takes part in other connections. These include several types of input to the olfactory bulb from central structures, such as feedback fibres and signal input fibres from brain stem systems.

Some of the neurons of the olfactory bulb transmit information from receptor cells to the brain. These neurons have a distinct spontaneous activity which is modified by olfactory stimulation. Some odours excite the olfactory neuron while others inhibit the activity. About 60% of the olfactory bulb units are inhibited, about 10% are unaffected and the rest are excited during olfactory stimulation. The tertiary structures of the olfactory system show even more pronounced inhibition than the bulb. The experimental data show that the receptors are frequently excited but seldom inhibited by a given stimulus. At the convergence point at the glomeruli of the olfactory bulb the activity is more differentiated. The neurons in the bulb have a strong tendency to be inhibited by odorous stimuli and specificity is indicated by differentially affected units. Most units are inhibited in the cortex.

2.1 Odorant-Receptor Interactions

In order to reach the olfactory receptors, an odour molecule in the air has to enter the aqueous environment of the epithelium. Thus the adsorption and desorption energies of the molecule from an air/lipoprotein interface will determine how easily a molecule will gain access to the

receptors. For odorant-receptor interaction to occur, the partition coefficients between water and lipoprotein will have some effect. The orientation of the molecule during its initial interaction with the receptor site will be influenced by electron donor /acceptor interactions which depend on the polarisability of the molecule. The molecular size and shape of the odorant molecule will determine the degree of fit with a particular site. A certain degree of affinity between the odorant and receptor site is necessary before olfactory transduction events can be triggered. Getchell (1986) and Lancet (1986) have reviewed the current understanding of the physiological and biochemical processes involved in olfactory transduction.

2.2 Receptor Specificity

Amoore (1967,1971) used psychophysical techniques to demonstrate the existence of specific anosmias (odour blindness) to several classes of odorants, and to quantify the extent of these anosmias in the human population. The number of these genetic defects of smell discovered have indicated that there are at least thirty different types of receptors. The existence of an olfactory defect may not preclude the detection of a particular odour class, even though that perception may be different, or require much higher concentrations before detection can occur. There is evidence that a given odour may interact with different receptor sites to different extents. This indicates that individual olfactory receptors are not tuned to individual odorants, but that they have a rather broad specificity. From the measurement of olfactory thresholds for different members of particular odorant classes, it is to be noted that some odorants can be perceived at very low concentrations. Thus 2-isobutyl-3-methoxy pyrazine (the smell of green peppers) can be perceived at 0.002 parts per billion (ppb). Yet side chain additions on the pyrazine ring can change the threshold enormously. Thus 2-isobutyl-3-methoxy-5,6-dimethylpyrazine can only be perceived at a concentration of 1,600 ppb. This large variation is likely to reflect the degree of fit of a particular molecule with one class of receptor site. The detection sensitivity will reflect the binding affinity of an odorant molecule for a particular receptor site. Molecules with some degree of flexibility may have the increased capability of binding with varying affinities to several classes of odorant receptors.

2.3 A hypothetical biological information processing algorithm for olfactory data

From the biological information we start with a collection of at least thirty types of odorant transducers which are characterized by an overlapping specificity toward different classes of volatile chemicals. This implies that the integrated response to a particular odour is the sum of the relative responses from the entire array of transducers. Thus an odorant stimulus X at a particular stimulus amplitude A can be described in terms of the response:

$$X_A = \sum_{i=0}^{i=N} x_i \quad (1)$$

where x represents the response to that particular concentration of odorant by each of N classes of odorant transducers. Such a concept has been discussed by Deutsch (1967).

The basis of many quantitative psychophysical measurements deals with the observation of the just noticeable difference (*jnd*) between two qualitatively similar stimuli. In all sensory modalities it is found that the *jnd* increases as the intensity of the stimulus increases and it is approximately proportional to the stimulus intensity I . This relationship, with some adjustment also applies to the olfactory sense, where the *jnd* falls between 0.21 and 0.41. This means that an odour stimulus must increase or decrease by 20-40% in order for us to detect a change, assuming that the stimulus is high enough to activate the receptor system but not to saturate the system. If the value of a parameter is represented to a neural processing element by the spike frequency in an input fibre, then processing errors in the element must be held below a *jnd* proportional to the parameter value. The widest effective range of operation of the processing element is obtained if the code for intensity of a stimulus is logarithmic. The physiological data from the vertebrate olfactory system suggest that this may be a valid assumption.

In order to distinguish between individual steps on a scale of intensity it is possible to describe a hypothetical criterion-indicating neuron. This incorporates a biasing inhibitory nerve current to a strong, slow decay, excitatory connection. The criterion frequency is the signal frequency for which the excitatory normalised connection current exceeds the inhibitory bias normalised connection current by the threshold current needed to cause the neuron to spike. A consequence is that the criterion-indicating neuron undergoes a transition from quiescence to spiking activity when the signal frequency exceeds a criterion value. Utilising several criterion indicating neurons, we can construct a neural analog-to-digital converter as shown in Fig. 2. Here the input signal fibre brings its logarithmic representation of the odour stimulus to a strong excitatory connection on each of N criterion indicating neurons arranged vertically in the figure. These neurons are identical except for their criterion values, which increase progressively from the bottom to the top of the column. This is done by letting all N excitatory signal connections be of the same strength and progressively increasing the strength of the inhibitory connection from the common source of fixed biasing nerve current. With the arrangement shown, nerve current in the single input fibre causes the neurons of sufficiently low column positions and hence low criterion values to spike. There is a sharp boundary of activity in the column with all neurons up to a particular position spiking rapidly and all other neurons quiet. The column height of the activity boundary is proportional to the input frequency and, therefore, to the logarithm of the parameter value. The range of values of the criteria of the N neurons is set to match the frequency range for which the input is not overloaded, but above its threshold, and N is made rather larger than the number of *jnd*'s

contained in the effective parameter range. With the limited intensity detection range of about $2 \log_{10}$ units in the olfactory system and a *jnd* of 20-40%, a value of 30 for N is adequate. At this point we have inputs from at least thirty different types of olfactory receptors, the intensity of the stimulus response from each receptor class being coded by a 31 bit logarithmic code. It must be stressed that we do not yet have enough information to say that such a process actually exists in the biological system, but the individual elements that have been identified can certainly be made to function in this way. Several alternative procedures may also be feasible.

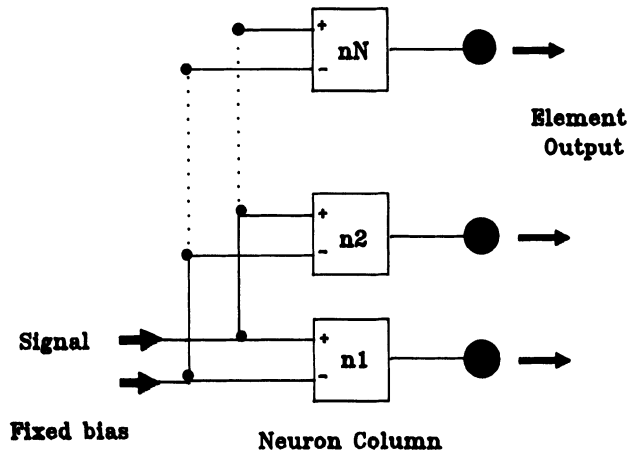


Figure 2 A neural analog to digital converter.

The elements of the circuit shown are criterion indicating neurons. These feature a biasing nerve current to a strong, slow decay, inhibitory connection. The signal consists of nerve current to a strong, slow-decay, excitatory connection. The criterion frequency of the neuron is the signal frequency for which the excitatory normalised connection current exceeds the inhibitory bias connection current by the threshold amount needed to cause the neuron to spike. The neuron column shown consists of a set of criterion indicating neurons which are identical except for their criterion values, which increase progressively from top to bottom. This occurs by letting all N excitatory signal connections be of the same strength and progressively increasing the strength of the inhibitory connection from the common source of fixed biasing nerve current. Looking at the element outputs of each neuron, the column height is proportional to the input frequency and under physiological conditions will normally be proportional to the logarithm of the signal input.

Olfactory discrimination is independent of the intensity of the stimulus within certain limits. Any model of olfactory discrimination must be stimulus amplitude independent. Thus a way might be found to transform the set of 30 odour intensity-dependent values to another set

of 30 values, of which only one is affected by the stimulus intensity. A possible algorithm which may be realised by known neuronal elements may be as follows:

- (a) find the highest of all the values of the 30 input parameters;
- (b) divide each of the 30 values by the highest value to form a ratio which becomes an odour descriptor;
- (c) weight these ratios to a digital code which are the output parameters of the system to be stored in memory.

The neural networks necessary for the realisation of such an algorithm have been discussed by Woolridge (1979), and a variation of this algorithm has been discussed by Deutsch (1967).

Establishment of the maximum parameter value can occur via a neuronal element known as a collecting neuron, whereby with 30 input channels from the neural analog-to-digital converters described, the output reflects the highest position of activity of all the inputs. Using an array of criterion gating neurons with the input to each element being divided into 30 equal groups biased by a fixed frequency representing the maximum intensity of the stimulus, it is possible to produce an output code which is a representation of the parameter ratios.

This model containing blocks of criterion-matching neurons can be reduced to the type of linear discriminant functions known as threshold elements, which are well characterised in the pattern recognition literature. A threshold element is a device with a binary output f related to the input variables $x_1, x_2, x_3, \dots, x_n$ by the equation:

$$f(x_1, x_2, x_3, \dots, x_n) = \begin{cases} 1 & \text{if } \sum_{i=1}^n w_i x_i \geq -w_{n+1} \\ 0 & \text{if } \sum_{i=1}^n w_i x_i < -w_{n+1} \end{cases} \quad (2)$$

where $-w_{n+1}$ is the threshold; $w_1, w_2, \dots, w_n$ are real numbers called the weights of inputs $x_1, x_2, \dots, x_n$, respectively. If the weights and threshold are fixed, $f(x_1, x_2, \dots, x_n)$ is a well defined and unique function, since every sample $x = x_1, x_2, \dots, x_n$ is mapped either to 0 or 1. This output corresponds to the two classes to be identified in a two class problem. If w is the hyperplane defined by $w = (w_1, w_2, \dots, w_{n+1})$ then the output of the threshold element is 0 if $w \cdot x < 0$ and 1 if x is on the positive side of the hyperplane. Individual threshold elements in the olfactory system may have variable thresholds. This would correspond to moving the hyperplane while maintaining its orientation, thus adjusting the boundary one way or the other between classes to be discriminated, and by so doing generate infinite numbers of subclasses. This would allow the subtle differences in the quality of an odour within a particular class to be detected.

A single threshold element will only distinguish between two categories. However, these can be placed in parallel multilayer networks which allow discrimination of a large number of variables.

3. REALISATION OF A PRACTICAL GAS SENSING DEVICE

It is clear from the above discussion that our knowledge of the olfactory system in vertebrates is far from complete, and that it is exceedingly complex. Nevertheless, it is possible to extract salient parts and form electronic analogies. In this section, we present recent developments in the design of gas and odour sensing apparatus.

3.1 Specification of Gas and Odorant Transducers

From our understanding of the olfactory system as outlined in Section 2, we can now map out the specifications of suitable transducers for gases and odours in an electronic system.

- (a) They do not need to be highly specific toward individual molecules, but should have some specificity toward particular classes of volatile chemicals.
- (b) Specificity can be achieved by arrays of such transducers, each with slightly different and overlapping response characteristics.
- (c) They should respond monotonically to the concentration of a volatile chemical in the environment.
- (d) It is desirable that they respond rapidly and reversibly at ambient temperature.
- (e) The signal produced by the transducer should be easily processed.

In our search for gas and odour transducers which fulfil these criteria we have developed devices based on electrically conducting organic polymers.

3.2 Conducting Polymer Gas and Odour Transducers

The best studied conducting polymer is polypyrrole. This is easily prepared, has high mechanical strength and good stability. Chemical methods for the oxidative preparation of polypyrrole from the pyrrole monomer are well known and reagents such as ferric chloride readily produce bulk polymer. However it was not until 1968 that Dall'Olio *et al.* established electrochemical methods of synthesis. Diaz *et al.* (1979) developed the methods of producing free-standing films by electrochemical methods and this has been steadily improved by other researchers.

In the last decade there has been a rapid expansion of research into highly conductive organic polymers. This new class of electronic material has a highly anisotropic quasi-one-

dimensional structure. In the conducting state, the material is ionic. Conductivity is higher along the polymer chain direction because of π - π electron overlap between successive monomers. Nevertheless, theoretical understanding of their electronic and structural properties lags behind experimental discoveries of practical applications of these polymers. This is largely because the polymers are black solids, insoluble in organic solvents and not amenable to conventional methods of analysis.

The electronic structure of organic compounds consists of:

- core electrons localised on a specific atomic nucleus;
- σ electrons localised between two bonded nuclei;
- n electrons associated with heteroatoms (O,N,S,P etc) which influence reactivity;
- π electrons which are involved in bonding but form weaker and less localised bonds which are less tightly bound than σ electrons.

The π electrons determine electron conduction in electroactive polymers. These electrons have a higher energy level than the σ electrons; thus there is an energy gap between the electrons in valence and conduction bands. The greatest stability of the conductance band occurs when atomic orbitals are parallel and orbital overlap between adjacent atoms is maximised. Fig. 3 shows some of the heterocyclic monomers commonly used.

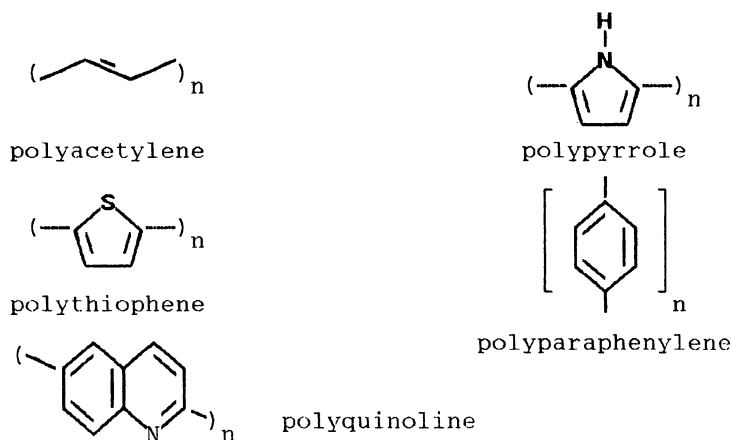


Figure 3 Heterocyclic monomers used for making conducting polymers.

Fig. 4 shows a space-filling model of hexapyrrole illustrating the quasi-one dimensional nature of the polymer and the orientation of monomeric units.

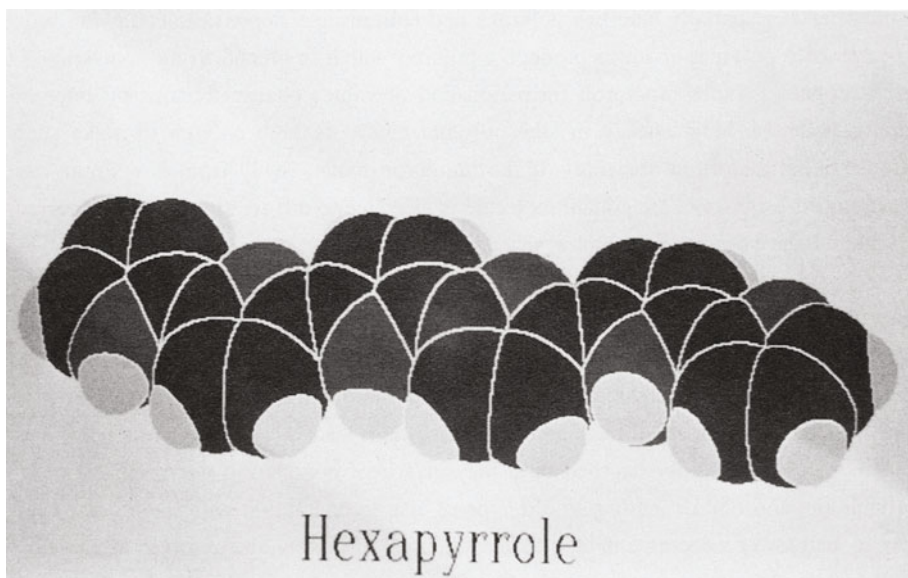


Figure 4 Space-filling model of hexapyrrole.

This consists of a nearly planar arrangement of pyrrole with one associated positive charge for every four monomeric units. It is thought that polar volatile molecules may act as dopants and thus modify the conductivity of the polymer transiently through formation of bipolaron charge carriers.

A basis for the charge carrying mechanisms within conducting polymers has now been established from theoretical work on model polymers and studies of the chemical and spectral properties of several conducting polymers (Scott *et al.* 1982, Street *et al.* 1982, Ogasawa *et al.* 1986, Cheung *et al.* 1988). The chain-like structure leads to strong coupling of the electronic states to conformational excitations (solitons, polarons and bipolarons) peculiar to one-dimensional systems. The relatively weak interchain binding allows diffusion of dopant molecules into the structure between the chains, while strong intra-chain bonds maintain the integrity of the polymer. Solitons have one electron in the mid-gap state between the valence and conduction band in energy, and may occur from transiently formed single bonds during electron delocalisation. Solitons can be neutral, positively or negatively charged. By themselves they have little effect on the conductivity of the polymer. Polarons (radical cations) can be formed by interaction between a charged and a neutral oliton to produce a charge defect. This radical cation can be stabilised by the presence of a dopant nion which counteracts the charge. Bipolarons (dications) can be formed by interaction of two polarons. The presence of bipolarons can induce large changes in conductivity since they can induce long range charge

the polymer can spatially stabilise polarons and solitons. A dopant molecule can induce ionisation of a polymer chain to produce a polaron which is pinned to the ionisation site. Further dopant induces bipolaron formation and promotes charge transfer by interchain hopping since the local energy in the polymer chain is high enough to make such a conformational distortion unstable. If the bipolaron moves away from the site at which ionisation was originated, the dopant molecule is then free to diffuse away from the polymer, or to again induce polaron formation at that particular site.

Persaud and Pelosi (1985, 1988) have investigated the gas sensing properties of over twenty conducting polymers. Reversible changes in the conductivity of polypyrrole to a variety of gases were observed. There was high sensitivity to polar molecules such as alcohols and amines and the change in conductivity was directly proportional to the concentration of vapour in air. There was little effect from non-polar molecules such as n-octane. It is feasible that volatile polar compounds adsorbed onto the surface of conducting polymers may act as reversible dopants and depending on the type of charge interaction with the polymer cause either an increase or a decrease in bipolaron formation. This would be observed as an increase or decrease in conductivity, respectively. Fig. 4 shows a model of a short polypyrrole chain, hexapyrrole. The polymer contains about one positive charge for every four monomeric units. Dopant counterions such as tetrafluoroborate or tosylate are often used. However, the dipole moment of a molecule such as ethanol adsorbed on to the polymer chain, is probably enough to pin down a positive charge and induce bipolaron formation. However, the exact mechanism involved in the conductance change is still unknown, and few tools exist for its investigation.

There has recently been much interest in the gas sensing properties of conducting polymers. Nylander *et al.* (1983), Miasik *et al.* (1986), Hanawa *et al.* (1988), Bartlett *et al.* (1989, 1990) have constructed sensors based on polypyrrole and investigated the response characteristics to ammonia and nitrogen dioxide and methanol.

In our research we have found that responses to gases are rapid and reversible at room temperature as shown in Fig 5. This is a unique characteristic of this type of gas sensor. The majority of gas transducers commercially available work at elevated temperatures between 60-500°C in order that adsorption and desorption of gaseous molecules occurs in reasonable time scales (seconds to minutes). We have observed with conducting polymer gas sensors that the response reaches a maximum within a few seconds and that recovery to baseline conditions, although dependent somewhat on the gaseous molecule detected, occurs within about 30 seconds at room temperature. It is likely that the initial interaction of a gaseous molecule with a polymer chain is by physical adsorption, but chemisorptive forces undoubtedly come in to play since the adsorbed molecule can also function as a dopant. Normally, chemisorptive processes are nearly irreversible phenomena. However, in this case where bipolaron probably formation is induced by the adsorbed molecule, the large local energy changes caused by

movement of the bipolarons away from the ionisation site probably facilitate desorption of volatile molecules adsorbed on to the polymer chain.

Response kinetics Methanol

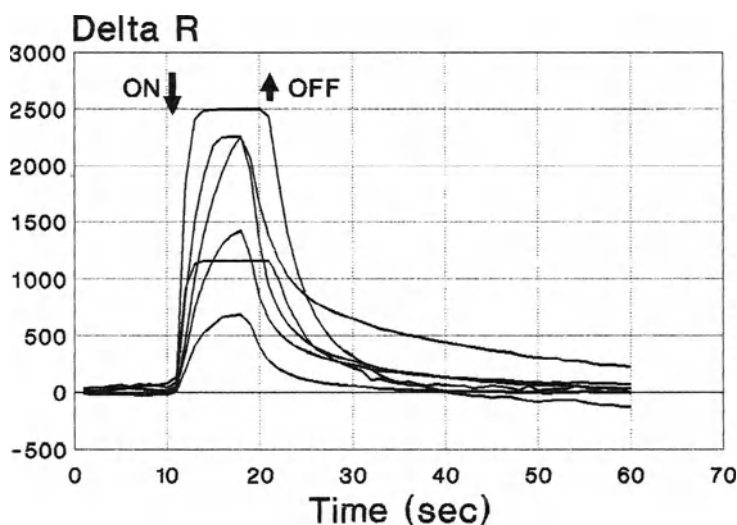


Figure 5 Response kinetics of polymer sensors.

Air containing methanol vapour was passed across the sensor array surface at 1 l/min. The responses are shown for six sensors in the array, demonstrating differing degrees of response from sensors made from chemically modified polymer elements.

The change in conductivity is proportional to the concentration of volatile molecule present. Fig. 6 shows the concentration response curve of a polypyrrole sensor to methanol. The Langmuir adsorption isotherm can be fitted to such a response curve. However, the available surface area is so large that it is difficult to achieve complete saturation of the sensor even at large concentrations of methanol. Hence, there is a large concentration range over which the concentration-response profile is nearly linear. This characteristic allows simple algorithms to be utilized for subsequent information processing.

For a given polymer, there are different degrees of response to different classes of volatile compounds at the same concentrations. Also the response of different conducting polymers to the same gas has been shown to be very different (Pelosi and Persaud, 1988).

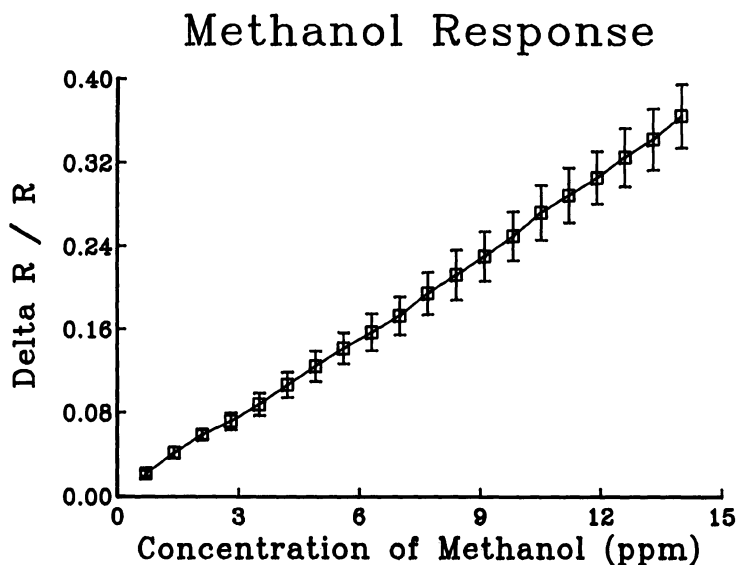


Figure 6 Concentration - response profile of a polypyrrole sensor to methanol.

The sensor was inserted into a 10 litre sealed glass vessel. Microlitre portions of methanol were injected through a septum into the vessel, and the system was allowed to equilibrate at room temperature. The change in sensor resistance for that concentration of vapour was recorded, following which a new portion of methanol injected.

It is now possible to compare the characteristics of conducting polymer gas and odour sensors with the specifications that we drew up in section 3.1. In terms of specificity, the organic polymers show a broad response spectrum, biased notably towards polar volatile compounds. The importance of molecular shape and size that is paramount in olfactory reception is less important in the interaction of volatile compounds with these polymers. However, the charge and dipole moment of the volatile molecule will be of great importance, as is also found in olfaction. Different polymers made from modified monomer units show broad overlapping specificities to different compounds. Hence, arrays of these sensors should behave very similarly to olfactory receptor arrays. The change in DC conductivity observed is easily measured, so that the responses from arrays of sensors can be easily collated. The chemistry of the monomers is sufficiently simple to allow tailoring of polymer specificity by making derivatives of the monomers containing hydrophobic chains or functional groups. However, it is not easily possible to tailor the molecular shape and size specificity for volatile compounds that is found in biological receptor systems.

4. A PRACTICAL GAS AND ODOUR SENSING DEVICE

We have been able to produce arrays of up to twenty gas sensitive polymers by a mixture of chemical and electrochemical synthesis using variations of techniques previously published (Diaz *et al.* (1988), Muhammadi *et al.* (1987)). The individual polymers are made from chemical derivatives of a variety of heterocyclic monomers chosen for their possible interaction with particular groups of volatile compounds. Such an array is shown in Fig. 7 where twenty sensors are incorporated into a standard 40 pin integrated circuit blank. The electronic circuit for continuous measurement of resistance of each sensor is built around a small microprocessor based unit shown in Fig.8. Essentially, a 12-bit digital-to-analog converter provides a programmable drive voltage which is sent via multiplexor switches to individual sensors. The current passing through that circuit is converted to a voltage which is digitized by a 12-bit analog-to-digital converter. Each individual sensor element can be interrogated at user defined intervals and the data stored in memory.

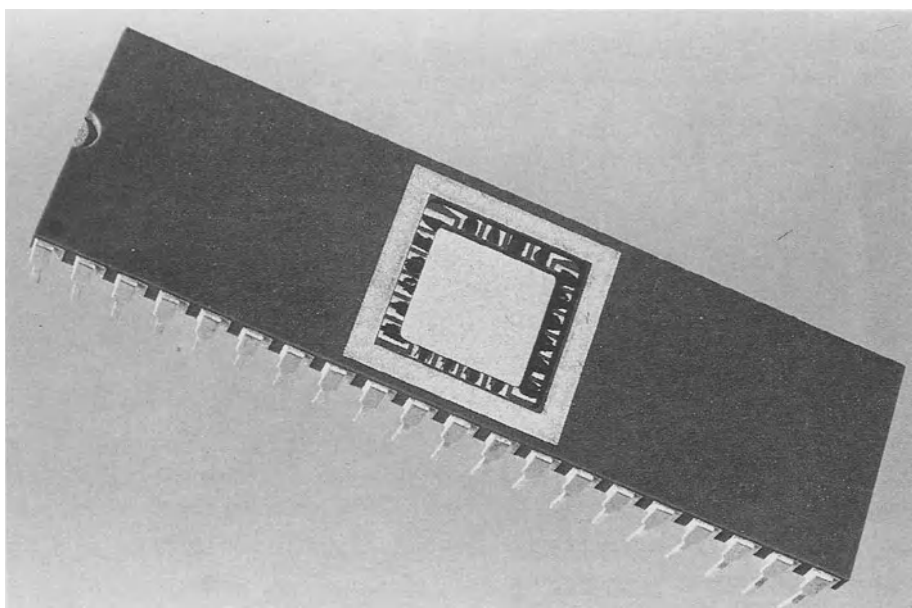


Figure 7 Array of twenty sensors.

The photograph shows 20 sensor elements formed on a standard 40 pin integrated circuit blank. The different polymers were grown on individual electrodes by a mixture of chemical and electrochemical techniques.

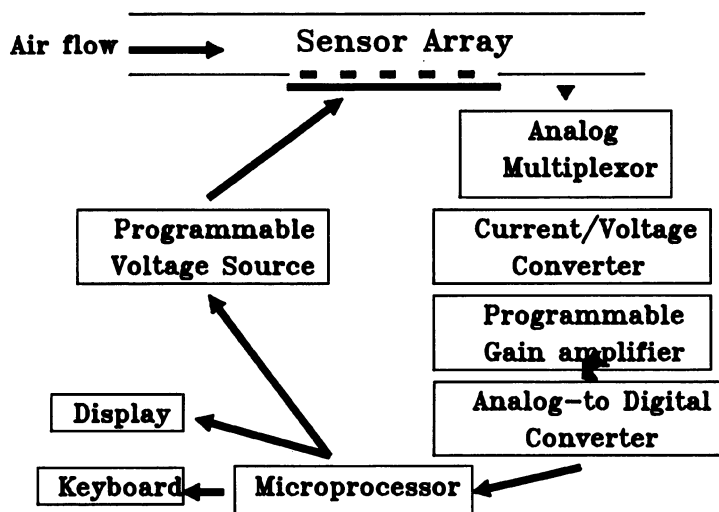
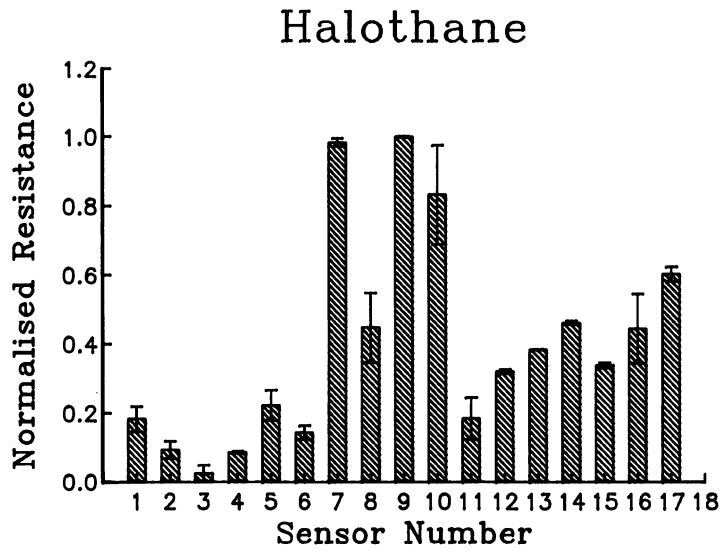


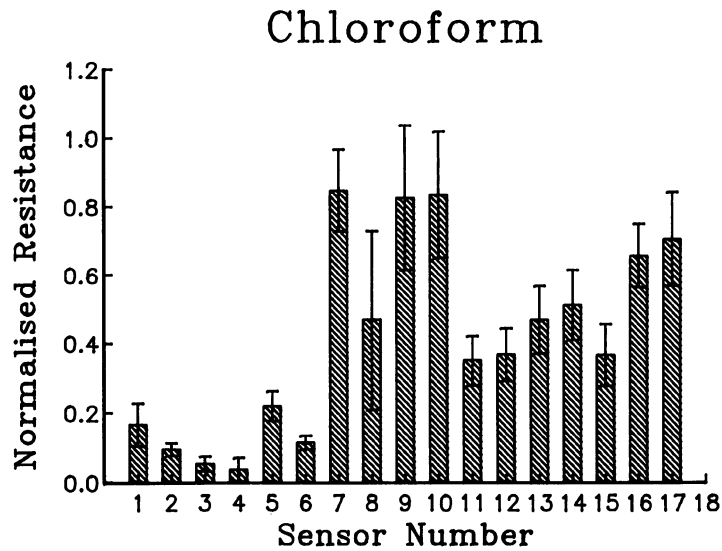
Figure 8 Electronic circuit for continuous measurement of sensor resistance.

Each sensor element changes in resistance when exposed to a volatile compound. However, the response to a given substance depends on the type of polymer element used so that a pattern of resistance changes can be recorded which is a set of descriptors for that gas. We have normalised the sensor responses in order to account for the differences in basal resistance of each sensor element and record the change in resistance/absolute resistance ($\Delta R/R$) for each sensor over the time profile of the volatile stimulus. This also produces a concentration independent pattern. Figs. 9(a-i) show the pattern of responses obtained for the following chemicals with an array of seventeen sensors: (a) halothane; (b) chloroform; (c) aniline; (d) pyridine; (e) methanol; (f) methoxyethanol; (g) ethyl acetate; (h) acetone; (i) water. For each sensor, the figures show standard deviations of the responses recorded over the peak of the stimulus profile. This gives a view of the inherent noise in the system, partly due to gas delivery to the polymer surface, as well as to some non-linearity of $\Delta R/R$ with concentration, so that local concentration changes of adsorbed gas can cause apparent noise. The concentration-response functions are being further investigated but, from Fig. 5, it is expected that variations over a wide concentration range should have no great effect on the overall pattern discrimination. The mean of several points are used and the standard deviation determines the confidence of discrimination between compounds.

Figure 9

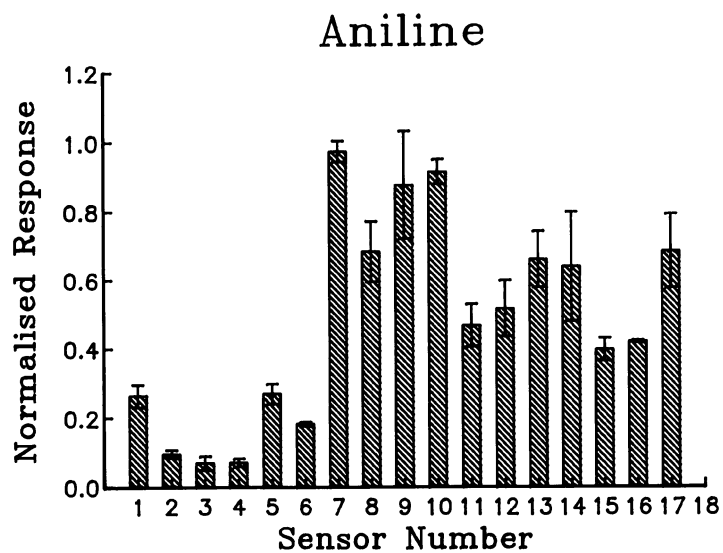


(a) Halothane

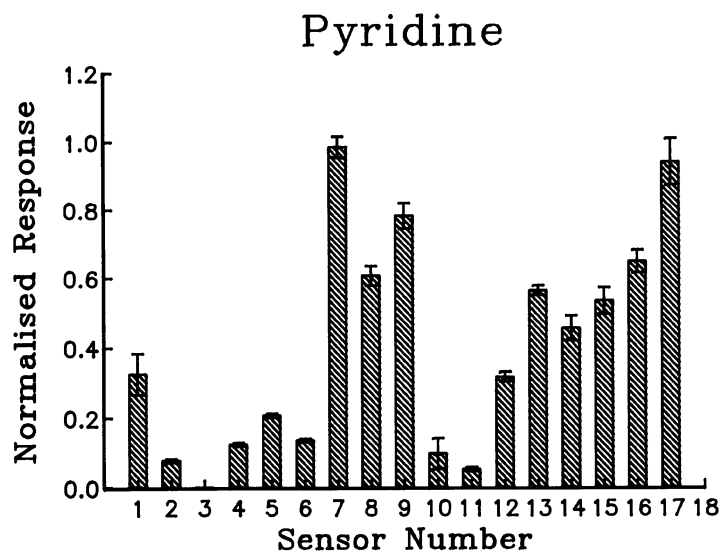


(b) Chloroform

Figure 9

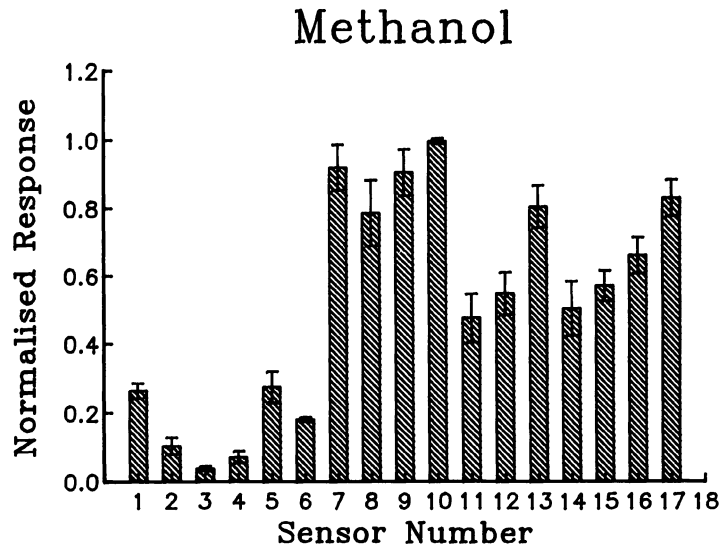


(c) Aniline

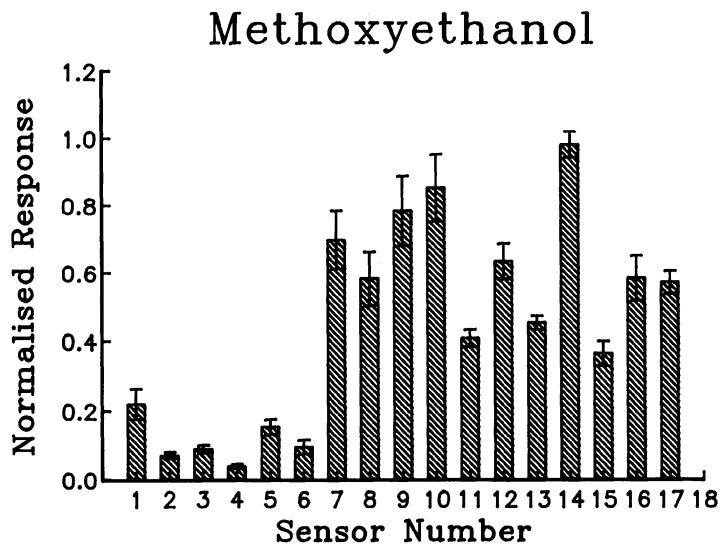


(d) Pyridine

Figure 9

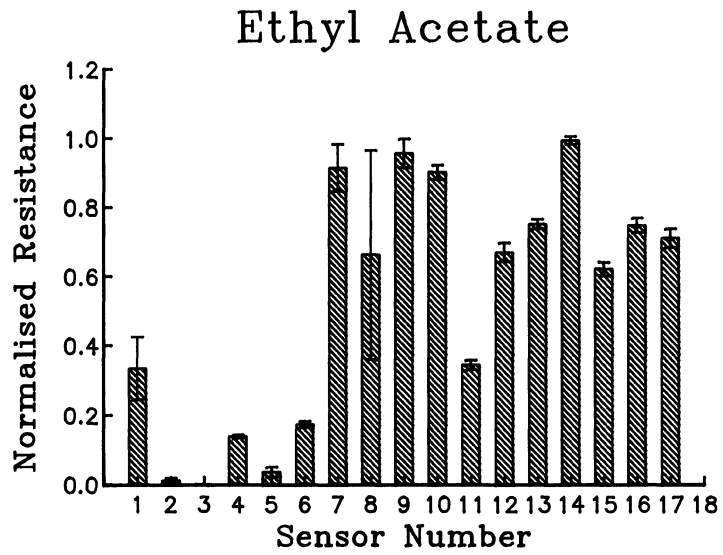


(e) Methanol

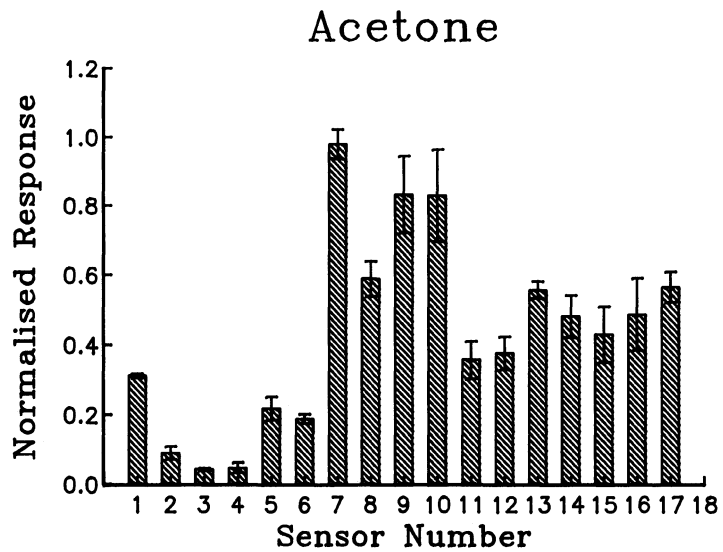


(f) Methoxyethanol

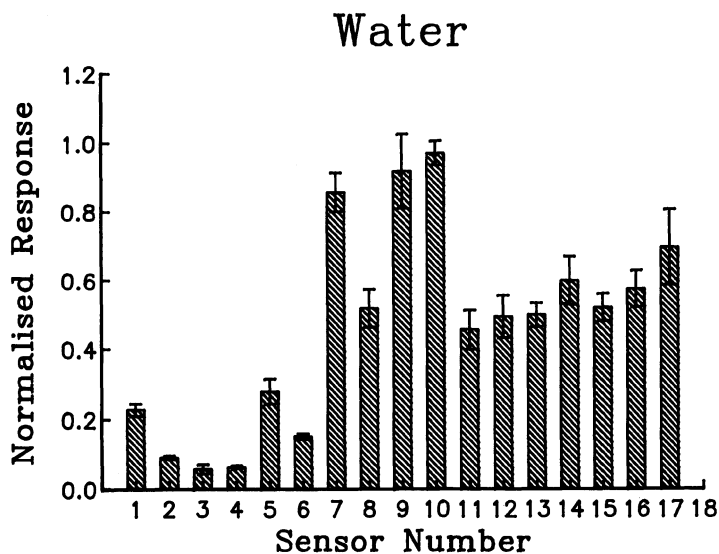
Figure 9



(g) Ethyl acetate



(h) Acetone



(i) Water

Figure 9 Normalised pattern of responses obtained by an array of seventeen sensors to different volatile chemicals.

The sensor array was presented with a ~10 sec pulse of vapour of each of the chemicals listed. The change in resistance of each sensor was recorded at intervals of 1.75 sec and four readings bracketing the peak response taken. These data were normalised and presented as a bar chart for each of seventeen sensors. The figure shows the pattern of responses obtained for each of the following compounds: (a) Halothane; (b) Chloroform; (c) Aniline; (d) Pyridine; (e) Methanol; (f) Methoxyethanol; (g) Ethyl Acetate; (h) Acetone; (i) Water

For Fig. 9, the following points should be noted:

- (i) Although individual sensors may respond to the same extent to different gases, other sensors respond to a greater or lesser extent to those gases, so that the overall pattern with the sensor array is unique to a particular gas.
- (ii) The pattern shown is independent of the concentration of gas. Thus the relative response to water vapour may be only a few percent of the response to methanol, yet a unique pattern may be observed.
- (iii) By also calibrating the sensor array with known concentrations of volatile compounds, both quality and intensity of a gas or odour can be determined.
- (iv) The pattern coding for a particular compound will be different from that in the olfactory system because of the different nature of the biological receptor sites, but it is similar in concept.

The data demonstrate the basis for a self learning gas and odour discrimination apparatus. The adaptive algorithms to be used are currently being developed. They are adaptations of the concepts outlined in equations 1 and 2. With seventeen sensors it is possible to determine the multidimensional space over which a particular gas can be defined to occupy, and to calculate its distance from another gas. This is shown in Fig. 10 where the difference in sensor responses between aniline and six other compounds are plotted. This graph describes how one compound can be distinguished from other compounds whose position in multi-dimensional space are known. Other results show that halothane can be distinguished from chloroform, but even more easily from pyridine which occupies an area of multi-dimensional space distant from halothane and chloroform.

Aniline

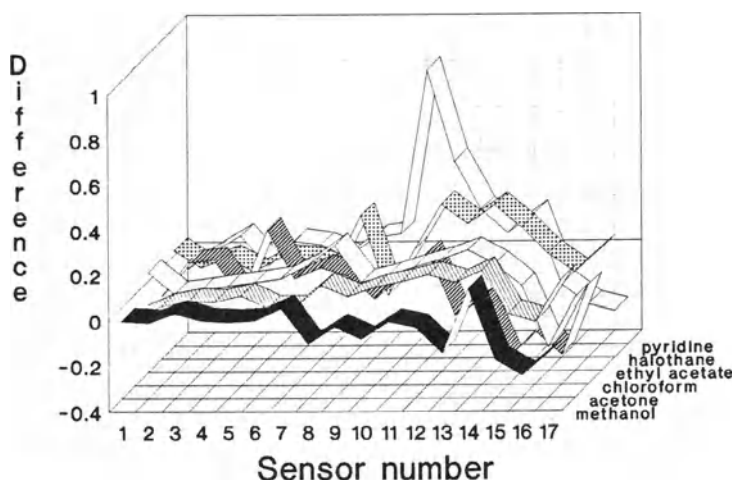


Figure 10 Discrimination of aniline from the volatiles shown in Fig. 9 Pyridine, Halothane, Ethyl Acetate, Chloroform, Acetone, Methanol

For aniline, the mean normalised response from each element of an array of seventeen sensors, was subtracted from that obtained for six other volatile chemicals. The results are plotted as a bar chart for each sensor and each chemical. It is clear that some sensors show a higher degree of specificity than others and these determine the distance of one volatile compound from another in multidimensional space. The diagrams illustrate the potential of the system for fine discrimination.

It is clear that some sensors show greater specificity toward certain compounds, and it is these that determine the distance between one gas and another in multi-dimensional space. Equation 2 describes the algorithm for distinguishing one gas from the other. This can be adapted to the data of Fig. 9 to determine the weighting factors necessary to describe the

hyperplane separating one compound from another in multidimensional space. Thus, given a known set of volatile compounds to be discriminated, the gas sensing system should maintain in the memory a table of molecular descriptors to distinguish one compound from another. Feature detection methods coupled to template-matching techniques have been shown to be successful. However, adaptive algorithms are necessary in order to compensate for changes in baselines or sensor response characteristics from day-to-day. Algorithms mimicking neural network processing such as those developed by Rumelhart (1986) are currently under investigation.

5. CONCLUSION

In the design of a multi-sensor apparatus for gas and odour discrimination, we have aimed at a flexible apparatus which can adapt to changing circumstances and needs of the users. The gas sensing elements are deliberately not specific to individual chemicals but respond with varying degrees to different chemical classes. However, each sensor has a different degree of specificity, although this may be very broad. As such, they resemble in concept the olfactory receptors found in the biological system. As can be seen from Figs. 9 and 10, the degree of pattern discrimination possible is potentially high. We are currently involved in further development of the chemistry of the sensing elements and the software necessary for pattern recognition.

ACKNOWLEDGEMENT - This work was financially supported by Cogent Ltd, Bucklersbury House, 3 Queen Victoria Street, London EC4N 8EL, UK.

REFERENCES

- Abe, H, Kanaya, S, Takahashi, Y and Sasaki, S. (1988) Extended studies of the automated odor-sensing system based on plural semiconductor gas sensors with computerized pattern recognition techniques. *Anal. Chim. Acta* 215, 155-168.
- Amoore, J.E. (1967) Specific anosmia: A clue to the olfactory code. *Nature*, 214, 1095-1098.
- Amoore, J.E. (1971) Stereochemical and vibrational theories of odour. *Nature*, 233, 270-271.
- Bartlett, P.N, Archer P.B.M and Ling-Chung, S.K. (1989) Conducting polymer gas sensors. Part I: fabrication and characterisation. *Sensors and Actuators* 19, 125-140.
- Bartlett, P.N and Ling-Chung, S.K. (1989) Conducting polymer gas sensors. Part II: response of polypyrrole to methanol vapour. *Sensors and Actuators* 19, 141-150.
- Cheung, K.M, Bloor, D and Stevens, G.C. (1988) Characterisation of polypyrrole electro-polymerised on different electrodes. *Polymer*, 29, 1709-1717.
- Deutsch, S. (1967) *Models of the Nervous System*. John Wiley, New York.
- Diaz, A.F, Kanazawa, K.K and Gardini, G.P. (1979) Electrochemical polymerisation of pyrrole. *J.Chem. Commun.* 635-636.

- Diaz, A.F, Robinson, J.F, and Mark, H.B. Jr. (1988) In: *Electronic Applications*, Springer-Verlag, Berlin, pp. 113-140.
- Dall'Olio, A, Dascola, V, Varacca, V and Bocche, V. (1968) Electron paramagnetic resonance and conductivity of an electrolytic oxypyrrole (pyrrole polymer) black. *Compte rend. l'Acad. Sci. (Paris)*, C267, 433.
- Dravnieks, A. and Trotter, P.J. (1965) Polar vapour detection based on thermal modulation of contact potential. *J. Sci. Instruments* 1, 102-119.
- Getchell, T.V. (1986) Functional properties of olfactory receptor neurons *Physiol. Rev.* 66, 772-818.
- Hanawa, T, Susumu, K and Yoneyama, H. (1988) Gas sensitivity of polypyrrole films to NO₂. *J. Chem. Soc. Faraday Trans.* 84, 1587-1592.
- Herberhold, C. (1969) Adsorption von Geruchsubstanzen auf Thermistoren In: *Verhandlungen der Deutschen Gesellschaft für Hals-Nasen-Ohren-Heilkunde, Kopf-und-Hals-Chirurgie auf der XL. Jahrsversammlung* (ed. H.H. Neumann) Springer-Verlag Berlin.
- Lancet, D. (1986) Vertebrate Olfactory Reception *Ann. Rev. Neurosci.* 9, 329-355
- Miasik, J.J, Hooper, A and Tofield, B.C. (1986) Conducting polymer gas sensors. *J. Chem. Soc. Faraday Trans.* 82, 1117-1126.
- McClelland, J.L, Rumelhart, D.E. and the PDP research group (1986) *Parallel Distributed Processing: Explorations in the microstructure of cognition. Vol.2: Psychological and Biological Models.* MIT Press. Cambridge, Mass.
- Muhammadi, A, Lundstrom, I, Salaneck, W.R. and Inganas, O. (1987) Polypyrrole prepared by chemical vapour phase deposition using hydrogen peroxide and hydrochloric acid. *Synth. Met* 21, 169-173.
- Nylander, C, Armgarth, M. and Lundstrom, I. (1983) An ammonia sensor based on a conducting polymer. *Anal. Chem. Symp. Series* 17, 203-207.
- Ogasawa, M, Funahashi, K, Demura, T, Hagiwara, T. and Iwata, K. (1986) Enhancement of electrical conductivity of polypyrrole by stretching *Synth. Met.* 14, 61-69.
- Persaud, K. and Dodd, G. (1982) Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose. *Nature* 299, 352-355.
- Persaud, K.C. and Pelosi, P. (1985) An approach to an artificial nose. *Trans. Am. Soc. Artif. Organs.* 31, 297-300.
- Pelosi, P. and Persaud, K.C. (1988) Gas Sensors: Towards an artificial nose. In: *Sensors and sensory systems for advanced robots* (ed. Dario, P.) NATO ASI Series: Series F: Computer and Systems Science, Springer-Verlag, Berlin. pp. 361-382.
- Scott, J.C, Pfluger, P, Clarke, T.C. and Street, G.B. (1982) Electron spin resonance and carbon-13-NMR studies of polypyrrole. *Polym. Prepr.* 23, 119.
- Street, G.B, Clarke, T.C, Krounbi, M.T, Kanazawa, K.K, Lee, V.Y, Pfluger, P, Scott, J.C. and Weiser, G. (1982) Preparation and characterization of neutral and oxidised polypyrrole films. *Mol. Cryst. Liq. Cryst.* 83, 1285-1296.
- Tanyolac, N.N, (1968) *Theories of odor and odor measurement* (Robert College Res. Center, Istanbul, Turkey).
- Woolridge, D.E. (1979) *Sensory Processing in the Brain: An Exercise in Neuroconnective Modelling.* John Wiley, New York.

Part 6

Interfacing Robots to Nervous System

MULTI-ELECTRODE STIMULATION OF MYELINATED NERVE FIBERS

P.H. Veltink, J.A. van Alsté, J. Holsheimer

Faculty of Electrical Engineering and
Coordination Center for Biomedical Technology,
University of Twente,
P.O. Box 217,
7500 AE Enschede,
The Netherlands.

INTRODUCTION

Patients with a central nervous system injury resulting in total or partial paralysis of extremities often have an intact peripheral neuromuscular system. Many attempts were made to restore lost functions by artificial electrical stimulation of the peripheral neuromuscular system. In principle, inducing muscle contraction is possible through electrical stimulation of:

- the ventral roots,
- the peripheral nerves or
- the muscles themselves.

In the latter case the nerve branches in the muscle are stimulated rather than the muscle fibers themselves (Grandjean and Mortimer, 1984; Eichhorn et al., 1984). The stimulation electrodes can be applied:

- on the surface of the skin (transcutaneous stimulation)
- through the skin (percutaneous stimulation)
- by implantation

Transcutaneous stimulation

Transcutaneous stimulation is the easiest to apply. It can be used for training atrophied muscles and temporary functional restoration of motion, but is not very well suited for long-term everyday use outside the clinic. Transcutaneous stimulation is clinically used for direct muscle stimulation as well as for nerve stimulation in relatively simple applications, like preventing drop foot by stimulating the peroneal nerve (Zilvold, 1976). Kralj et al. (1983) of the Ljubljana research group were among

the first to investigate the application of transcutaneous stimulation for gait restoration in selected complete paraplegic patients. Using their multi-channel stimulation system they achieved simple biped gait for up to 20 to 30 minutes with the assistance of parallel bars or a roller-walker. The reproducibility of net muscle force is highly variable in many applications of transcutaneous stimulation. This is especially true for direct muscle stimulation, because of large geometrical deformations between the skin electrode and excitable nerve branches due to muscular contraction.

Percutaneous stimulation

Percutaneous stimulation was developed by Peckham et al. (1980) in Cleveland for stimulating arm muscles, and was also used in the lower extremities (Marsolais and Kobetic, 1983, 1986 and 1987). Thin coiled wires were inserted through the skin into the muscle via a hypodermic needle. Good muscle selectivity can be achieved, but stimulation characteristics are muscle length dependent and electrode leads may break (Peckham et al., 1987). Furthermore, care must be taken to prevent infection at the places where the electrode leads cross the skin.

Implantable devices

Implantable devices for stimulation of the neuromuscular system have been developed both for direct muscle stimulation and/or nerve stimulation. Hildebrandt and Meyerwaarden (1984) developed small stimulator-electrode configurations which can be implanted into the muscle. Recently, Smith et al. (1987) from the Cleveland group realized an implantable system for direct muscle stimulation via electrodes on the surface of the muscle (epimysial electrodes). Grandjean and Mortimer, 1986, showed in acute animal experiments the dependence of recruitment properties on muscle length when using epimysial electrodes. This length dependence was minimal for a monopolar electrode close to the nerve entrance. This position also gave the best muscle selectivity and highest slope of the recruitment curve. In chronic experiments they found that after 30 days the epimysial electrode was encapsulated by fibrous tissue and was incorporated into the fascial layer and thus removed from the optimal position. The muscle length dependence was now slightly larger and the muscle selectivity had become lower. The functional characteristics of implantable muscle stimulation systems are comparable with the percutaneous method.

The application of implantable systems for electrical stimulation of the neuromuscular system in paralyzed patients with an injured central nervous system has been tested clinically (e.g. for restoring gait) but is still in an experimental stage. In most clinical trials the peripheral nerves were stimulated. Advantages are muscle length independence, the possibility of stimulating several muscles from one position, fewer mechanical problems, and low energy consumption. However, muscle selectivity is generally better with direct muscle stimulation. Peroneal nerve stimulation for correction of footdrop was investigated by Waters et al. (1984) and Acimovic-Janezic et al. (1984). A problem encountered with this application was the changing balance between inversion and eversion of the foot during dorsiflexion, a reason for McNeal and Bowman (1984) to investigate muscle selective nerve stimulation with multi-electrode configurations. Femoral nerve stimulation was applied in a patient by Zilvold et al. (1984). Holle et al. (1984) used a 16-channel nerve stimulation system for standing and walking in complete paraplegic patients. The functional performance of their system is not significantly better than the performance described by Kralj et al. (1983) for their transcutaneous stimulation system.

The stimulation system must be easy to apply and preferably be "forgettable" to the patient in the case of long term use in a home situation. This can be achieved by implanting both the electrodes and the stimulator. Nerve stimulation is preferred in implantable systems, except when nerves are not accessible or the required muscle selectivity cannot be achieved. It is commonly found in clinical practice that paraplegic patients and others will not accept the implantation of a nerve stimulation system unless functions are restored that cannot be restored by the use of orthoses or wheelchairs alone. At the moment this condition is not fulfilled. This is mainly due to inadequate 1) control strategies, and 2) nerve stimulation methods. Control strategies will not be dealt with in this paper, although it has our research interest (Van Alsté et al. 1987,1988, Mulder et al. 1987, Willemsen et al. 1986, Veltink et al., 1989c, 1989d). Problems with nerve stimulation methods include inefficient muscle contraction, fast muscle fatigue, poor control of muscle force and poor muscle selectivity. Improved nerve stimulation methods may be expected to result from further research in this area, leading to clinically acceptable results.

MOTOR UNIT RECRUITMENT IN HEALTHY SUBJECTS

Contractions of artificially stimulated muscles are different from those under physiological control and are not optimal. Although muscles can accommodate to the way they are used (plasticity of muscle: Peckham et al. 1976; Edgerton and Cremer, 1981; Eerbeek et al. 1984), muscles contract more efficiently when the artificially induced contractions follow more physiological patterns. This is true for firing times as for motor unit recruitment. Thus, less muscle fatigue and a better grading of muscle force would occur (Holle et al., 1974; Petrofsky, 1978, 1979; Solomonow, 1983; Fang and Mortimer, 1987). It is therefore essential to study physiological contraction control in normal subjects.

A motor unit consists of all the muscle fibers innervated by one motoneuron. All fibers in a motor unit have the same histochemical and physiological characteristics (Kugelberg, 1981). Motor unit characteristics are determined by the fiber number and the properties of the constituent muscle fibers that vary within a muscle from small slowly contracting units to larger, fast contracting units. The slow units are fairly fatigue resistant and generally have a smaller force contribution. Fast contracting units vary between smaller fatigue resistant and larger, fast fatiguable units (Burke et al., 1973, 1981a,b). The contribution of a motor unit to the force a muscle exerts (its "size") is related to the diameter of the innervating motoneuron: larger sized units have thicker motoneurons (Burke et al., 1973; Kernell, 1986).

During physiologically controlled muscle contractions, motor units contract asynchronously. The central nervous system is organized so as to recruit each motor unit above a certain contraction level at which the muscle is already exerting. This level depends on the size of the motor unit. Larger units are recruited at a higher threshold (Henneman's size principle: Henneman et al., 1965a,b, 1981; Kernell 1986). The mean frequency at which a motor unit is fired above its threshold increases with the contraction level (Henneman et al., 1965b). The physiological control strategy is correlated with the motor unit composition of a muscle, which differs strongly between muscles. This recruitment strategy, built into the central nervous system, is denoted by a physiological recruitment order. A central problem of artificial stimulation is that artificial recruitment order can deviate appreciably from the physiological order. Also, asynchronous contraction of motor units is replaced by synchronized contraction.

ARCHITECTURE OF PERIPHERAL NERVES

The architecture and composition of nerves and nerve fibers influence the recruitment of nerve fibers during artificial nerve stimulation.

Peripheral nerves consist of nerve fiber bundles called fascicles (figure 1). These bundles are surrounded by connective tissue of rather inhomogeneous composition (Sunderland, 1968). The structure is denser both immediately around the fascicles (perineurium) and at the surface of the nerve as compared to the bulk of the connective tissue sheath (epineurium). The epineurium also contains structures such as blood vessels. The number of fascicles in a nerve and the amount of connective tissue varies considerably both between different nerves in the same individual and between corresponding nerves in different species. Sunderland has described the architecture of peripheral nerves in man (1968).

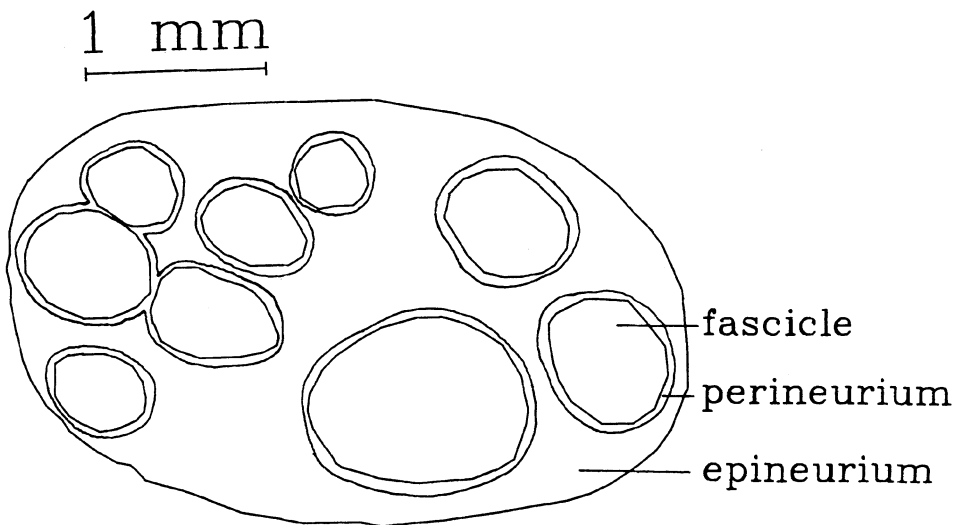


Figure 1
Multi-fascicular human nerve: example of a cross-section of deep peroneal nerve, 24 mm below the neck of the fibula, as described by Sunderland (1968).

Fascicles frequently split up and merge on their way from the central nervous system to the periphery (Sunderland, 1968). Over a distance of 10 mm, the distribution of fascicles over the cross section of a nerve can change completely.

This fascicular plexus inside the nerve can differ between individuals and results in a redistribution of nerve fibers in the nerve. At small distances from the central nervous system nerve fibers in peripheral nerves intermingle with nerve fibers of other muscles. More distally, they are localized in fascicles that are muscle specific and eventually branch to the muscle. This complex architecture has to be considered when several muscles are to be stimulated selectively.

Boyd and Davey (1968) investigated the composition of peripheral nerves in the cat: The diameters of myelinated nerve fibers in peripheral muscle nerves vary between 2 and 18 μm . Neurons efferently innervating the motor units in the muscles are called alpha motoneurons. Their axons have diameters varying between 10 and 17 μm and constitute about 30 % of the myelinated nerve fibers in peripheral muscle nerves. Other efferent nerve fibers in motor nerves are the fusimotor fibers innervating the muscle spindles (nerve fiber diameters between 2 and 9 μm). Approximately 40% of the nerve fibers in a motor nerve are afferent fibers. They transport information from the periphery (sensory endings of muscle spindles, tendon organs, free nerve endings for sensing pressure or pain, blood vessel nerve endings, etc.) to the central nervous system. The diameter of these afferent fibers varies between 1 and 17 μm . At the roots of the spinal cord motoneuron fibers and myelinated sensory fibers are divided between ventral and dorsal roots, respectively (Clark, 1984).

All of the above mentioned nerve fibers are myelinated: they are surrounded by electrically isolating myelin produced by Schwann cells that are spirally wrapped around the axon. Unmyelinated nerve fibers are usually present in peripheral nerves in great numbers (Thomas and Ochoa, 1984), but their diameters are much smaller (less than 3 μm) and they have much higher thresholds for stimulation (Ranck, 1975). In myelinated nerve fibers the ratio between axon diameter and total fiber diameter (including myelin sheath) is about 0.7 (Thomas and Ochoa, 1984). The isolating myeline sheath is interrupted at regular intervals at the nodes of Ranvier. These nodes are spaced at longer intervals on larger than on smaller fibers. The internodal length is roughly proportional to the diameter of the nerve fiber and ranges from about 0.1 mm on the smallest fibers to 1.8 mm on the largest fibers (Thomas and Ochoa, 1984).

Electrical currents between the intracellular and interstitial media mainly flow at the nodes of Ranvier, because only at these nodes is the nerve fiber membrane in direct contact with the interstitial tissue. The distribution of the nodes plays an important role in determining which fibers will be activated by artificial nerve stimulation.

UNPHYSIOLOGICAL CHARACTERISTICS OF ARTIFICIAL NERVE STIMULATION SYSTEMS

As described, nerve stimulation for the restoration of lost functions of paralyzed extremities has rarely been used in clinical practice so far, because the results of clinical trials were not satisfactory. Main bottle necks were the lack of control strategies and insufficient methods for nerve fiber stimulation.

In the classical method of artificial nerve stimulation the nerve is stimulated bipolarly using an isolating cuff with two ring electrodes surrounding the nerve. To prevent mechanical damage to the nerve the diameter of the cuff needs to be a few times the nerve diameter. Recently, Naples et al., 1988 introduced a spiral nerve cuff electrode with a 'self-sizing' property. This electrode has the advantage of fitting to peripheral nerves snugly without damaging them due to tissue ingrowth.

Blair and Erlanger (1933) showed an inverse order of recruitment as compared to the physiological situation during extraneural stimulation of frog sciatic nerve. The same was found by Petrofsky (1978) and Fang and Mortimer (1987) by measuring the widths of single twitch muscle contractions. Petrofsky stimulated the ventral roots of cats using extraneural bipolar ring electrodes. Fang and Mortimer stimulated a branch of the sciatic nerve in cats via a bipolar cuff configuration. Recruitment of large motor units before small units, as is the case with inverse recruitment, leads to faster muscle fatigue. The synchronized excitation of the same motor units also leads to faster muscle fatigue. In order to obtain smooth muscle contractions the stimulation frequency had to be quite high as compared to the physiological frequency of motor unit recruitment.

Further adverse characteristics of this nerve stimulation approach are: a limited possibility for muscle contraction grading by recruitment modulation due to the steep relation between modulated stimulation parameters and muscle force (recruitment curve), a limited range for frequency modulation of muscle contraction and no possibilities for independent control of contraction of different muscles or muscle parts innervated by the same stimulated nerve.

Several approaches were taken to obtain a more physiological way of controlling muscle contraction. Methods for restoring physiological recruitment order were proposed. Furthermore, multi-electrode configurations were introduced in order to stimulate motor unit groups alternately, which leads to smoother muscle contractions at lower motor unit contraction frequencies. Also, multi-electrode configurations could lead to independent contraction control of different muscles with the

same electrode configuration. Experimental research on nerve stimulation was complemented by theoretical studies using simulation models.

EXPERIMENTAL STUDIES ON NERVE STIMULATION

Several methods were investigated for obtaining a **physiological order of nerve fiber recruitment** in order to improve fatigue resistance. They are all based on the idea that if a neuron unintentionally tends to fire too early with respect to the physiological order, it could also be blocked early. Petrofsky used a constant DC anodal blocking voltage (Petrofsky, 1978; Petrofsky and Phillips, 1981), which cannot be used in long term applications. Solomonow investigated a high-frequency blocking technique (Solomonow et al., 1983a,b, 1984). He used two bipolar cuff electrodes. The first electrode excited all nerve fibers and controlled the contraction rate. The second electrode controlled motor unit recruitment by blocking part of the recruited nerve fibers by high frequency stimulation. Fang and Mortimer (1987) realized the same function with one tripolar electrode configuration consisting of a cathode for exciting motoneurons and two anodes for blocking some of the largest motoneurons recruited.

The above authors all showed improvements in fatigue resistance during isometrical muscle contraction (Petrofsky and Phillips, 1981; Solomonow et al., 1983b; Fang and Mortimer, 1987). The blocking methods were all tested in configurations where only one electrode determined the stimulus rate of the recruited motoneurons. It would be an interesting idea to combine these blocking methods with multi-electrode configurations to also enable the alternating contraction of motor unit groups and the independent contraction control of different muscles or muscle parts.

Van den Honert and Mortimer (1981a,b) introduced another blocking method for blocking unwanted action potentials generated in the spinal cord. They developed an asymmetrical tripolar cuff electrode configuration for generating unidirectionally propagating action potentials. Blocking was obtained by collision of these antidromic action potentials with action potentials generated by another electrode configuration placed proximally to the tripolar cuff. Sweeney and Mortimer (1986) developed and tested an asymmetric two electrode cuff for the same application.

The application of multi-electrode configurations for alternating stimulation of muscle parts to reduce muscle fatigue was introduced by Thoma and Holle for

long-term phrenic nerve stimulation for assisting respiration (Holle et al., 1974). An improved resistance to fatigue was found in dogs (Holle et al., 1974) as well as in humans (Moritz, 1983). Multielectrode configurations were also used in a functional stimulation system for the lower limbs (Thoma et al., 1978, Holle et al., 1984). Small electrodes were sewn along the epineurium of the nerve. Alternating recruitment of motor unit groups was obtained by varying electrode combinations. The electrode combination was changed after each muscle contraction, not after each stimulation pulse. Although they used a relatively complex stimulation system, the functional results in restoring walking in paraplegic patients were not significantly better than in surface stimulation systems. In our opinion an important reason was the lack of a good control strategy. Other reasons were the absence of recruitment modulation for each electrode combination (each stimulated muscle could only be turned on or off), and the limited number of muscles which were stimulated.

Petrofsky (1979) investigated alternating stimulation in cat sciatic nerves. A cuff configuration with three stimulation electrodes was used. Each electrode was positioned at one side of the nerve. At each stimulation pulse one electrode was stimulated and the other two were connected to the reference voltage. The active electrode was changed at each stimulation pulse. By introducing additional grounding electrodes he could obtain almost the same fatigue resistance as was found when splitting up the associated motoneurons at the ventral roots into three parts and stimulating them separately. He did not, however, compare these results with fatigue resistance obtained with a one-electrode cuff configuration. Although combining stimulation and reference electrodes in one cross-section of the nerve may change the constitution of the group of motoneurons that are recruited, the stimulation current required will be considerably higher than when the reference electrode is placed further along the nerve, because the potential drop along the nerve is the determining factor in the excitation of motoneurons (Rushton, 1927; Ranck, 1975).

Possibilities for muscle specific stimulation using a seven-electrode cuff configuration around the sciatic nerve in dogs were investigated by McNeal (McNeal, 1985). The electrode configuration consisted of one reference electrode and six electrodes positioned at two sides along the nerve. McNeal used bipolar stimulation of two electrodes on the same side along the nerve and performed electromyography (EMG) with wire electrodes in four leg muscles. Two of these muscles are innervated by the posterial tibial nerve, the other two by the peroneal nerve. Fascicles branching into each of these nerves lie at either side of the sciatic nerve at the place of stimulation. Muscle selective stimulation with the seven-electrode configu-

ration could be obtained when electrodes were in direct contact with the nerve and positioned correctly. Talonen (1985) was also able to selectively control the contraction of different muscles by stimulating the sciatic nerve of cats using a four electrode configuration.

Solomonow et al. investigated optimal schemes for recruitment and frequency modulation during nerve stimulation (Zhou et al., 1987; Solomonow et al., 1987). Their aim was to obtain linear relations between the stimulation input signal, muscle force, and mean absolute EMG (to be used in closed loop control of muscle contraction). The optimal recruitment and frequency modulation strategies depended on the predominant fiber composition of the stimulated muscle.

Studies covering areas other than functional aspects of nerve stimulation have been investigated, but will not be discussed in detail here. The requirements for limiting electrode corrosion and tissue damage during nerve stimulation were investigated by Donaldson et al. (Donaldson and Donaldson, 1986; Craggs et al., 1986). Limited morphological changes in nerves during chronic nerve stimulation were reported by several investigators (Breederveld and Den Otter, 1981; Bowman and Erickson, 1985). Even during the chronic application of coiled wire intraneural electrodes, only minimal nerve damage and connective tissue growth around the electrodes were observed (Bowman and Erickson, 1985).

MODEL STUDIES ON NERVE STIMULATION

Modelling nerve stimulation on the basis of experimental results can result in a quantitative description of the influence of such parameters as nerve fiber diameters, the location of the nodes of Ranvier, volume conduction characteristics, and electrode positions on nerve fiber recruitment. The knowledge resulting from such studies can be useful in explaining experimental results and for designing new electrode configurations.

Most nerve stimulation models described in the literature use the concept introduced by McNeal (1976). Using his model, McNeal determined strength-duration curves with a chronaxie of 80 μ s, which is consistent with data reported by BeMent and Ranck (1969) for stimulation of myelinated nerve fibers in the dorsal column of the cat. Furthermore, the model predicts an increase in threshold with decreasing fiber diameter and, thus, inverse recruitment order. This was found experimentally for extraneural electrodes (Petrofsky, 1978; Fang and Mortimer, 1987).

Some authors incorporated the influence of current redistribution between extracellular and intracellular media under subthreshold conditions in the volume conduction model by assuming the nerve fascicles have an anisotropic conductivity (Plonsey, 1974; Stegeman, 1979, 1982) or by using the bidomain approach (Plonsey and Barr, 1987). Clark and Plonsey (1970) theoretically described the influence of an active unmyelinated nerve fiber on an inactive unmyelinated fiber in the same nerve fascicle.

Volume conduction characteristics can influence nerve fiber excitation to a great extent. McNeal (1979) used a homogeneous isotropic volume conduction model. Altman and Plonsey (1986) and Ferguson et al. (1987) used an inhomogeneous model in order to theoretically study the performance of asymmetric tripolar and bipolar electrode cuffs for the unidirectional propagation of action potentials. Such nerve stimulation configurations were experimentally tested by Van den Honert and Mortimer (1981a,b) and Sweeney and Mortimer (1986).

McNeal (1978) evaluated nerve fiber excitation within a cuff electrode for two node positions. He found that node position can significantly influence the excitation thresholds of nerve fibers, especially for stimulation electrodes and nerve fibers in close proximity. McNeal (1977) showed in a theoretical model that cathodal monopolar stimulation excitation will first occur at the node nearest to the electrode.

The McNeal model was verified experimentally by Gorman and Mortimer (1983) with respect to the influence of stimulation pulse parameters. Reduction of the width of the primary cathodic stimulation pulse and the use of a secondary anodic pulse for charge balancing both reduced the slope of the relationship between stimulus amplitude and evoked muscle force (recruitment curve). These effects were also predicted by the model simulations, as was an increased difference of the excitation threshold for fibers of different diameters.

Rattay (1986) proposed the second order difference of the externally applied electrical potential along a myelinated nerve fiber as an indication for depolarization and hyperpolarization regions. This second order difference is the input term of each partial differential equation from the set of coupled equations used by McNeal to describe nerve fiber excitation on the basis of his network model. Although this approach can not be used to determine whether nerve fibers are

excited or not, it gives a good impression of the regions where nerve fibers may be excited.

Few nerve stimulation models incorporated a simple inhomogeneous volume conduction model (Altman and Plonsey, 1986; Ferguson et al., 1987). For a good understanding of nerve stimulation it is important, however, to investigate the influence of a realistic volume conductor on the recruitment of nerve fibers. Furthermore, the distribution of the discrete positions of the nodes of Ranvier, the geometrical fiber distribution inside the nerve and the nerve fiber diameter distribution influence nerve fiber recruitment, but none have been investigated theoretically. Experimental studies to verify the results of these investigations must then be performed.

RESEARCH AT THE UNIVERSITY OF TWENTE ON NERVE STIMULATION

Alternating stimulation of groups of motor units belonging to a single muscle or a number of synergistic muscles can reduce muscle fatigue if the overlap of the groups is minimal. Furthermore, the contraction of muscles having different functions can be controlled independently, if each of these muscles can exclusively be activated by at least one of the stimulating electrodes. It would be advantageous if independent control of several muscles could be accomplished by stimulating a nerve at one position using a multi-electrode configuration.

Most human nerves have many fascicles with different muscle specificities. With extraneural stimulation muscle selectivity can be achieved when fascicles innervating the same muscle are concentrated on the same side of the nerve (McNeal, 1985). Muscle selective stimulation would improve if it were possible to selectively stimulate each fascicle. Electrodes in the epineurium just outside each fascicle or intrafascicular electrodes may be needed to selectively stimulate fascicles. It is therefore important to investigate the recruitment of nerve fibers inside a fascicle by electrodes close to or in the fascicle. We did research directed towards this aim.

Nerve fiber excitation on a fascicular level was tested experimentally in the rat common peroneal nerve with extraneural and intrafascicular electrodes (Veltink et al. 1989). This nerve was chosen because it consists of one main fascicle, sometimes with an additional smaller one, surrounded by little connective tissue. For intrafascicular electrodes a mixed order of recruitment could be demonstrated. Furthermore, it appeared difficult to recruit non-overlapping motoneuron groups within one

fascicle with different electrodes placed in the same cross-section outside or inside the fascicle.

The positions of the nodes of Ranvier are important in determining excitation thresholds of nerve fibers if a stimulating electrode is in or near a fascicle. In this case the extracellular potentials along the nerve fibers vary substantially over small distances as compared to the internodal length of the nerve fibers. Node positions, nerve fiber positions and nerve fiber diameters are not known. However, probability distributions of these three parameters may be estimated. Veltink et al. (1988a) gave a probabilistic description of nerve fiber excitation as a function of fiber position and stimulation pulse amplitude. For given nerve fiber diameters, these excitation probabilities were calculated using a mathematical simulation model. This model incorporated a volume conduction model which consisted of two cylinders and represented a mono-fascicular nerve. Excitation probabilities were calculated for two electrode configurations: an intrafascicular point and an extraneural ring electrode.

The influence of the volume conduction model parameters on the excitation probability distributions were analyzed by Veltink et al. (1988b). An inverse recruitment in a probabilistic sense was predicted in most cases; excitation probability increases with nerve fiber diameter at given stimulus pulse amplitudes. The extent of the inverse recruitment depended on volume conduction parameters and electrode position. The results correspond with the mixed recruitment found by intrafascicular stimulation of the rat common peroneal nerve (Veltink et al. 1989) and with pronounced inverse recruitment for extraneural stimulation (Blair and Erlanger, 1933, Petrofsky, 1978, Fang and Mortimer, 1987).

Nerve fiber excitation in the rat common peroneal nerve was simulated in a mathematical model incorporating realistic nerve and electrode geometries (Veltink et al. 1989b). Experimental results in multi-electrode stimulation of one fascicle, suggesting extensive overlap, could thus be explained by the influence of node positions. Furthermore, the possibility of fascicle selective stimulation was investigated in a mathematical model of a multi-fascicular nerve for extraneural, epineural, and intrafascicular stimulation. The simulation results were compared to the experimental results of McNeal (1985) in the case of extraneural stimulation and to our experimental results for epineural and intrafascicular stimulation. The model predicted a good fascicle selectivity for electrodes placed in the fascicle. For electrodes placed just outside a fascicle, neighboring fascicles will also be stimulated (Figure 2). Fascicle selectivity increases when the electrode is placed closer to the fascicle.

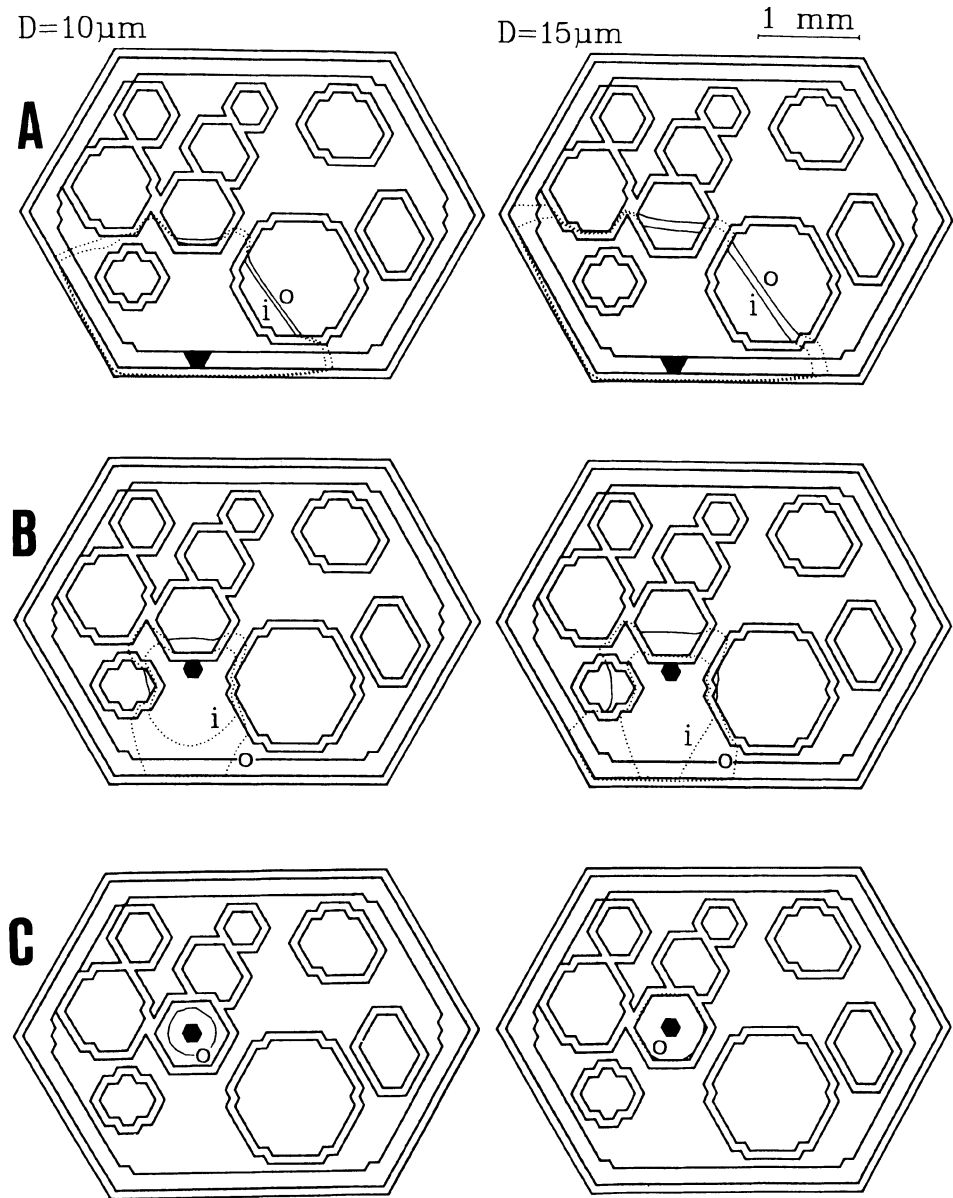


Figure 2

Simulated nerve fiber recruitment contours for the example cross-section of human deep peroneal nerve (figure 1) (Veltink et al., 1989b).

The contour lines *i* and *o* surround the areas in which the model predicts nerve fibers to be excited for fixed nerve fiber diameter *D*. All nerve fibers will be excited inside the inner contour *i* and none will be excited outside the outer contour *o*. In the area between both contours nerve fiber excitation depends on node positions.

A : extraneural electrode.

B : epineural electrode, placed just outside the perineurium of a fascicle.

C : intrafascicular electrode.

Conclusions from our research are:

- The application of multi-electrode configurations for the stimulation of one fascicle will be difficult if electrodes are placed in one nerve cross-section in or near the fascicle. This is due to the great influence of discrete node positions, which results in rather extensive overlap of motoneuron groups recruited with different electrodes.
- A mixed rather than a strict inverse order of recruitment was theoretically predicted and experimentally found with small distances between electrodes and nerve fibers.
- Theoretical studies predicted that the selectivity of multi-electrode configurations in multi-fascicular nerves may be expected to improve when fascicles are selectively stimulated by electrodes in or near the fascicles. This has to be verified experimentally.
- A technological approach for realizing an electrode configuration could be: Distribution of many electrodes outside or inside a nerve along a distance of several cm. The electrodes should preferably be in direct contact with the nerve, where the amount of artificial (synthetic) material should be minimized. Intra-neural electrodes should be optimally located inside different fascicles from a functional point of view. However, in cases when this would damage the fascicles, the electrodes should be placed outside the fascicles in direct contact with the perineurium. In this case, it may be necessary to isolate the electrodes from the surrounding epineural tissue. The electrodes used during functional stimulation should be selected based on their stimulus responses. The selection procedure should be repeated with time.

Research in the area of nerve stimulation is being continued at the University of Twente. Koole et al. (1989) reported on fascicle selective stimulation. Rutten et al. (1988) and Meier et al. (1989) reported on nerve fiber selective stimulation, and Struijk et al. (1988) investigated spinal cord stimulation.

REFERENCES

- Acimovic-Janezic, R., Krajnik, J., Simic, V., Strojnik, P., Implantable Peroneal underknee stimulator - evaluation, Proc. 8th Int. Symposium on External Control of Human Extremities, Dubrovnik, 1984, pp. 487-492.
- Alsté, J.A. van, Veltink, P.H., Nijmeijer, H., Muscle length control by electrical nerve stimulation. Proc. 9th An. Conf. IEEE/Eng. Med. & Biol. Boston, 1987, Vol. 3, pp. 1372-1373.
- Alsté, J.A. van, Brug, H. ten, Mulder, A.J., Stand up with functional electrostimulation, Int. Series on Biomechanics, Vol 7-A, Free University Press, Amsterdam, 1988, pp 214-219.
- Altman, K.W., Plonsey, R., A Two-Part Model for Determining the electromagnetic and Physiologic Behavior of Cuff Electrode Nerve Stimulators, IEEE Trans. Biomed. Eng., Vol. 33, 1986, pp. 285-293.

- BeMent, S.L., Ranck, J.B., A Quantitative Study of Electrical Stimulation of Central Myelinated Fibers, *Experimental Neurology*, Vol. 24, 1969, pp. 147-170.
- Bowman, B.R., Erickson, R.C., Acute and chronic implantation of coiled wire intraneural electrodes during cyclical electrical stimulation, *Ann. Biomed. Eng.*, Vol. 13, 1985, pp. 75-93.
- Boyd, I.A., Davey, M.R., *Composition of Peripheral Nerves*, E & S Livingstone, Edinburgh and London, 1968.
- Breederveld, R.S., Otter, G. den, Electrical stimulation of the Femoral Nerve in the cat, *Proc. 7th International Symposium on External Control of Human Extremities*, Dubrovnik, 1981, pp. 137-138.
- Burke, R.E., Levine, D.N., Tsairis, P., Zajac, F.E., Physiological Types and Histochemical profiles in motor units of the cat gastrocnemius, *J. Physiol.*, Vol. 234, 1973, pp. 723-748.
- Burke, R.E., Motor Units: anatomy, physiology, and functional organization, in: *Handbook of Physiology*, Vol. 2, Am. Physiol. Soc., 1981a, pp. 345-422.
- Burke, R.E., Motor Unit Recruitment: What Are the Critical Factors?, in: J.E. Desmedt (Ed.), *Motor Unit Types, Recruitment and Plasticity in Health and Disease*, *Prog. clin. Neurophysiol.*, Vol. 9, 1981b, pp. 61-84.
- Clark, J.W., Plonsey, R., A Mathematical study of nerve fiber interaction, *Biophysical Journal*, Vol. 10, 1970, pp. 937-957.
- Clark, R.G., Anatomy of the mammalian cord, in: Davidoff, R.A. (ed.), *Handbook of the Spinal Cord*, Vol. 2 and 3, Dekker, New York, 1984, pp. 1-45.
- Craggs, M.D., Donaldson, N. de N., Donaldson, P.E.K., Performance of platinum stimulating electrodes, mapped on the limit-voltage plane. Part 1: Charge injection in vivo, *Med. & Biol. Eng. & Comput.*, vol. 24, 1986, pp. 424-430.
- Donaldson, N. de N., Donaldson, P.E.K., Performance of platinum stimulating electrodes mapped on the limit-voltage plane. Part 2: Corrosion in vitro, *Med. & Biol. Eng. & Comput.*, Vol. 24, 1986, pp. 431-438.
- Edgerton, V.R., Cremer, S., Motor Unit Plasticity and Possible Mechanisms, in: J.E. Desmedt (Ed.), *Motor Unit Types, Recruitment and Plasticity in Health and Disease*, *Prog. clin. Neurophysiol.*, Vol. 9, 1981, pp. 220-240.
- Eerbeek, O., Kernell, D., Verhey, B.A., Effects of fast and slow patterns of tonic long-term stimulation on contractile properties of fast muscle in the cat, *J. Physiol.*, vol. 352, 1984, pp. 73-90.
- Eichhorn, K.Fr., Schubert, W., David, E., Special problems in the stimulation of denervated muscles, *Proc. 8th Int. Symposium on External Control of Human Extremities*, Dubrovnik, 1984, pp. 137-145.
- Fang, Z.P., Mortimer, J.T., A method for attaining natural recruitment order in artificially activated muscles, *IEEE 9th Annual Conference of the Engineering in Medicine and Biology Society*, Vol. 2, 1987, pp. 657-658.
- Ferguson, A.S., Sweeney, J.D., Durand, D., Mortimer, J.T., Finite difference modelling of nerve cuff electric fields, *IEEE 9th Annual Conference of the Engineering in Medicine and Biology Society*, Vol. 3, 1987, pp. 1579-1580.
- Frankenhaeuser, B.H., Huxley, A.F., The action potential in the myelinated nerve fiber of *Xenopus laevis* as computed on the basis of voltage clamp data, *J. Physiol.*, Vol. 171, 1964, pp. 302-315.
- Gorman P.H., Mortimer, J.T., The Effect of stimulus Parameters on the Recruitment Characteristics of Direct Nerve Stimulation, *IEEE Trans. Biomed. Eng.*, Vol. 30, 1983, pp. 407-414.
- Grandjean, P.A., Mortimer, J.T., Recruitment properties of monopolar and bipolar epimysial electrodes, *Proc. 8th Int. Symposium on External Control of Human Extremities*, Dubrovnik, 1984, pp. 453-464.
- Grandjean, P.A., Mortimer, J.T. Recruitment properties of monopolar and bipolar epimysial electrodes. *Annals of Biomed. Engin.* Vol. 14, 1986, pp. 53-66.
- Henneman, E., Somjen, G., Carpenter, D.O., Functional significance of cell size in spinal motoneurons, *J. Neurophysiol.*, Vol. 28, 1965a, pp. 560-580.
- Henneman, E., Somjen, G., Carpenter, D.O., Excitability and inhibibility of motoneurons of different sizes, *J. Neurophysiol.*, Vol. 28, 1965b, pp. 599-619.

- Henneman, E., Recruitment of Motoneurons: the Size Principle, in: J.E. Desmedt (Ed.), *Motor Unit Types, Recruitment and Plasticity in Health and Disease*, Prog. clin. Neurophysiol., Vol. 9, 1981, pp. 26-60.
- Hildebrandt, J.J., Meyer-Waarden, K., Development of an EMG-controlled 8-channel system for neuromuscular functional stimulation, Proc. 8th Int. Symposium on External Control of Human Extremities, Dubrovnik, 1984, pp. 443-452.
- Holle, J., Moritz, E., Thoma, H., Lischka, A., Die Karussellstimulation, eine neue Methode zur elektrophrenische Langzeitbeatmung, Wiener klinische Wochenschrift, Vol. 86, 1974, pp. 23-27.
- Holle, J., Frey, M., Gruber, H., Kern, H., St hr, H., Thoma, H., Functional Electrostimulation of Paraplegics, Experimental Investigations and First Clinical Experience with an Implantable Stimulation Device, Orthopedics, Vol. 7, 1984, pp. 1145-1155
- Kernell, D., Organization and properties of spinal motoneurons and motor units, Progress in Brain Research, Vol. 64, 1986, pp. 21-30.
- Koole, P., Put J.H.M., Veltink P.H., Holsheimer J., Muscle selective stimulation for FES, Proc. 3rd Vienna Int. Workshop on FES, Vienna, 1989, pp.155-158.
- Kralj, A., Bajd, T., Turk, R., Krajnik, J., Benko, H., Gait Restoration in Paraplegic Patients: A Feasibility Demonstration using Multichannel Surface Electrode FES, J. of Rehabilitation R&D, Vol. 20, 1983, pp. 3-19.
- Kugelberg, E., The Motor Unit: Morphology and Function, in: Desmedt (ed.), *Motor Unit Types, Recruitment and Plasticity in Health and Disease*, Karger, Basel, 1981, pp. 1-16.
- Marsolais, E.B., Kobetic, R., Functional Walking in Paralyzed Patients by Means of Electrical Stimulation, Clinical Orthopaedics, 1983, Vol. 175, 1983, pp. 30-36.
- Marsolais, E.B., Kobetic, R., Implantation techniques and experience with percutaneous intramuscular electrodes in the lower extremities, Journal of Rehabilitation Research and Development, Vol. 23, 1986, pp. 1-8.
- Marsolais, E.B., Kobetic, R., Functional Electrical Stimulation for Walking in Paraplegia, The Journal of Bone and Joint Surgery, Vol. 69A, 1987, pp. 728-733.
- Meier J.H., Rutten W.L.C., Put J.H.M., Fibre selective neural stimulation using intrafascicular electrodes, Proc. 11th An. Conf. IEEE/Eng. Med & Biol., Seattle, 1989, Vol 3, pp. 988-989.
- McNeal, D.R., Analysis of a Model for Excitation of Myelinated Nerve, IEEE Trans. Biomed. Eng., Vol. 23, 1976, pp. 329-337.
- McNeal, D.R., Teicher, D.A., Effect of Electrode Placement on Threshold and initial site of Excitation of a Myelinated Nerve Fibre, in: Hambrecht, F.T. and Reswick, J.B. (eds.), *Functional electrical stimulation*, New York, Marcel Dekker, 1977, pp. 405-412.
- McNeal, D.R., Apkarian, A.V., Analytical Design of a Cuff Electrode, Proc. 6th Int. Symposium on External Control of Human Extremities, Dubrovnik, 1978, pp. 45-54.
- McNeal, D.R., Bowman, B.R., Selective activation of muscles using peripheral nerve electrodes, Med. & Biol. Eng. & Comput., Vol. 23, 1985, pp. 249-253
- Moritz, E., Holle, J., Thoma, H., St hr, H., Further experiences with elektrophrenic respiration, Proc. 1st Vienna International workshop on Functional Electrostimulation, 1983, 6.5.
- Mulder, A.J., Alsté, J.A. van, Hermens, H.J., Zilvold, G., Closed loop control of standing up. Proc. top EC COMAC BME workshop on Restoration of walking aided by functional electrical stimulation. Enschede, 1987, pp. 37-46.
- Naples, G.G., Mortimer, J.T., Scheiner, A., Sweeney, J.D., A Spiral Nerve Cuff Electrode for Peripheral Nerve Stimulation, IEEE Trans. on Biomed. Eng., Vol. 35, 1988, pp. 897-904.
- Peckham, P.H., Mortimer, J.T., Marsolais, E.B., Alteration in the Force and Fatigability of Skeletal Muscle in Quadriplegic Humans Following Exercise Induced by Chronic Electrical Stimulation, Clinical Orthopaedics and Related Research, Vol. 114, 1976, pp. 326-334.
- Peckham, P.H., Marsolais, E.B., Mortimer, J.T., Restoration of key grip and release in the C6 tetraplegic patient through functional electrical stimulation, The Journal of Hand Surgery, Vol. 5, 1980, pp. 462-469.
- Peckham, P.H., Keith, M.W., Thrope, G.B., Memberg, W.D., Intramuscular electrodes for neuromuscular stimulation, IEEE 9th Annual Conference of the Engineering in Medicine and Biology Society, Vol.3, 1987, pp. 1569-1570.
- Petrofsky, J.S., Control of the recruitment and firing frequencies of motor units in electrically stimulated muscles in the cat, Med. & Biol. Eng. & Comput., Vol. 16, 1978, pp. 302-308.

- Petrofsky, J.S., Sequential motor unit stimulation through peripheral motor nerves in the cat, *Med. & Biol. Eng. & Comput.*, Vol. 17, 1979, pp. 87-93
- Petrofsky, J.S., Phillips, C.A., Impact of recruitment order on electrode design for neural prosthetics of skeletal muscle, *Am. J. Phys. Med.*, Vol. 60, 1981, pp. 243-253.
- Plonsey, R., The Active Fiber in a Volume Conductor, *IEEE Trans. Biomed. Eng.*, Vol. 21, 1974, pp. 371-381.
- Plonsey, R., Barr, R.C., Interstitial potentials and their change with depth into cardiac tissue, *Biophys. J.*, Vol. 51, 1987, pp. 547-555.
- Ranck, J.B., Which elements are excited in electrical stimulation of mammalian central nervous system: a review, *Brain Research*, Vol. 98, 1975, pp. 417-440.
- Rattay, F., Analysis of Models for External Stimulation of Axons, *IEEE Trans. Biomed. Eng.*, Vol. 33, 1986, pp. 974-977.
- Rushton, W.A.H., The effect upon the threshold for nervous excitation of the length of nerve exposed, and the angle between current and nerve, *J. Physiol. (London)*, Vol. 63, 1927, pp. 357-377.
- Rutten W.L.C., Wier H.J. van, Put J.H.M., Rutgers R., Vos R.A.I. de, Sensitivity, Selectivity and Bioacceptance of an intraneural multi-electrode stimulation device in silicon technology, *Proc. 7th Congress of ISEK, Enschede, Excerpta Medica*, 1988, pp. 135-139.
- Smith, B., Peckham, P.H., Gazdik, M., Letechipial, J.E., Banks, S.A., Keith, M.W., Development and evaluation of an externally powered multichannel, implantable stimulator, *IEEE 9th Annual Conference of the Engineering in Medicine and Biology Society*, Vol.2, 1987, pp. 622-623.
- Solomonow, M., Eldred, E., Lyman, J., Foster, J., Control of muscle contractile force through indirect high-frequency stimulation, *Am. J. Phys. Med.*, Vol. 62, 1983a, pp. 71-82.
- Solomonow, M., Eldred, E., Lyman, J., Foster, J., Fatigue considerations of muscle contractile force during high-frequency stimulation, *Am. J. Phys. Med.*, Vol. 62, 1983b, pp. 117-122.
- Solomonow, M., External Control of the Neuromuscular system, *IEEE Trans. Biomed. Eng.*, Vol. 31, 1984, pp. 752-763.
- Solomonow, M., Baratta, R., Zhou, B.H., Shoji, H., D'Ambrosia, R.D., The EMG-Force Model of Electrically Stimulate Muscle: Dependence on Control Strategy and Predominant Fiber composition, *IEEE Trans. Biomed. Eng.*, Vol 34, 1987, pp. 692-703.
- Stegeman, D.F., De Weerd, J.P.C., Eijkman, E.G.J., A Volume Conductor study of compound action potentials of nerves in situ: the forward problem, *Biol. Cybernetics*, Vol. 33, 1979, pp. 97-111.
- Stegeman, D.F., De Weerd, J.P.C., Modelling compound action potentials of peripheral nerves in situ. I. Model description; evidence for a non-linear relation between fiber diameter and velocity, *Electroenceph. Clin. Neurophysiol.*, Vol. 54, 1982, pp. 436-448.
- Strojnjk, P., Vavken, E., Zerovnik, S., Simic, V., Stopar, M., Stanic, U., Acimovic-Janezic, R., Implantable Peroneal underknee stimula-tor, *Proc. 8th Int. Symposium on External Control of Human Extremities*, Dubrovnik, 1984, pp. 479-485.
- Struijk, J.J., Holsheimer J., Veen B.K. van, Analysis of Dorsal Column Stimulation, *Proc. 10th An. Conf. IEEE/Eng. Med. & Biol.*, New Orleans, 1988, Vol.4, pp. 1692-1693.
- Sunderland, S., Nerves and Nerve Injuries, E & S Livingstone, Edinburgh and London, 1968.
- Sweeney, J.D., Mortimer, J.T., An Asymmetric Two Electrode Cuff for Generation of Unidirectionally Propagated Action Potentials, *IEEE Trans. Biomed. Eng.*, Vol. 33, 1986, pp. 541-549.
- Sweeney, J.D., Mortimer, J.T., Durand, D., Modeling of mammalian myelinated nerve for functional neuromuscular stimulation, *IEEE 9th Annual Conference of the Engineering in Medicine and Biology Society*, Vol. 3, 1987, pp. 1577-1578.
- Talonen, P., Baer, G., Huhti, M., Hakkinen, V., Control of muscle force by sequential motor unit stimulation of peripheral nerves, Control of muscle force by sequential motor unit stimulation of peripheral nerves, *Proc. XIV Int. Conf., on Med. & Biol. Eng. and VII Int. Conf. on Med. Physics*, 1985, pp. 396-397.
- Thoma, H., Holle, J., Moritz, E., Stoehr, H., Walking after paraplegia - a principle concept, *Proc. 6th Int. Symposium on External Control of Human Extremities*, Dubrovnik, 1978, pp. 71-80

- Thomas, P.K., Ochoa, J., Microscopic Anatomy of Peripheral Nerve Fibers, in: P.J. Dyck, P.K. Thomas, E.H. Lambert, R. Bunge (Eds.), *Peripheral Neuropathy*, Vol. 1, W.B. Saunders, Philadelphia, 1984, pp. 39-96.
- Van den Honert, C., Mortimer, J.T., A Technique for Collision Block of Peripheral Nerve: Single Stimulus analysis, *IEEE Trans. Biomed. Eng.*, Vol. 28, 1981a, pp. 373-378.
- Van den Honert, C., Mortimer, J.T., A Technique for Collision Block of Peripheral Nerve: Frequency Dependence, *IEEE Trans. Biomed. Eng.*, Vol. 28, 1981b, pp. 379-382.
- Veltink P.H., Alsté, J.A. van, Boom, H.B.K., Simulation of Intrafascicular and extraneural nerve stimulation. *IEEE Trans. Biomed. Eng.* Vol. 35, 1988a, pp. 69-75.
- Veltink, P.H., Alsté, J.A. van, Boom, H.B.K., Influences of stimulation conditions on recruitment of myelinated nerve fibers: A model study. *IEEE Trans. Biomed. Eng.* Vol. 35, 1988b, pp 917-924.
- Veltink, P.H., Alsté, J.A. van, Boom, H.B.K., Multi-electrode Intra- fascicular and extraneural stimulation. *Med. & Biol. Eng. & Comput.*, Vol. 27, 1989, pp 19-24.
- Veltink, P.H., Veen, B.J. van, Struijk, J.J., Holsheimer, J., Boom, H.B.K., A model study on fascicle selective nerve stimulation. Accepted for publication by *IEEE Trans. Biomed. Eng.* 1989b.
- Veltink, P.H., Alsté J.A. van, Dynamics and control of a muscle-load system during electrical nerve stimulation, *Automedica*, Vol. 11, 1989c, pp. 91-98.
- Veltink, P.H., El-Bialy A., Chizeck H.J., Crago P.E., Nonlinear control of an artificially stimulated muscle-skeleton-load system, *Proc. 11th An. Conf. IEEE/Eng. Med. & Biol.*, Seattle, Vol. 3, 1989d, pp. 969-970.
- Waters, R.L., McNeal, D.R., Clifford, B., Faloona, W., Long-term follow-up of peroneal NMA patients, *Proc. 8th Int. Symposium on External Control of Human Extremities*, Dubrovnik, 1984, pp. 471-477.
- Willemsen, A.T.M., Alsté, J.A. van, Boom, H.B.K., Measuring FES feedback parameters with accelerometers. *Proc. 8th An. Conf. IEEE/Eng. Med. & Biol.* Dallas, 1986, pp. 671-674.
- Zhou, B.H., Baratta, R., Solomonow, M., Manipulation of Muscle Force with Various Firing Rate and Recruitment Control Strategies, *IEEE Trans. Biomed. Eng.*, Vol. 34, 1987, pp. 128-139.
- Zilvold, G., *Functionele elektrostimulatie van de n. Peroneus*, Doctor Thesis, Free University, Amsterdam, 1976.
- Zilvold, G., Hermens, H.J., Kreuwel, H., Waal, C. de, Alsté, J.A. van, Experiences with an implantable Nervus Femoralis stimulator, *Proc. 8th Int. Symposium on External Control of Human Extremities*, Dubrovnik, 1984, pp. 465-469.

THE ROLE OF MATERIALS IN DESIGNING NERVE GUIDANCE CHANNELS AND CHRONIC NEURAL INTERFACES

Robert F. Valentini and Patrick Aebischer
Section for Artificial Organs, Biomaterials and Cellular Technology
Brown University
Providence, RI 02912 USA

Abstract:

A variety of neural electrodes have been developed to simulate and record from axons of the peripheral nervous system. Chronic neural interfaces may be used to facilitate voluntary movement or to control prosthetic devices in patients with spinal cord injuries, amputated appendages, or damaged nerves. Severed nerves are secured within synthetic tubular structures so as to induce and guide regenerating axons through neural interfaces. The regeneration process is strongly influenced by the properties of the tubular guidance channels. Channel characteristics impacting on regeneration include electrical, permeability, and surface microgeometrical properties as well as the ability to release growth factors. Biocompatible electrode materials are essential for insuring axonal survival and accurate signal transduction, especially in long-term use. By maximizing regeneration and providing a compatible neural interface, it may be possible to design implantable devices capable of chronic neural signal transduction in humans. Prosthetic devices and control units connected to neural interfaces could restore motor and sensory function. This review describes current approaches toward the development of nerve guidance channels and chronic neural interfaces.

Introduction

The potential of the peripheral nervous system to regenerate in man has been appreciated since the latter part of the nineteenth century, although the regulatory mechanisms are still unknown (1). Numerous biological and synthetic adjuncts have been used to facilitate repair and enhance regeneration. Many of the earlier attempts failed because of infection, mechanical failure, or intense tissue reaction to the implanted materials. In the 1970's, a variety of novel synthetic polymers displaying good biocompatibility and structural integrity became available. This factor, combined with advances in microsurgical techniques and the elucidation of basic processes modulating neural regeneration, opened new possibilities for peripheral nerve repair (2).

Over the past decades electrodes to stimulate or record from neural tissue have also been developed. Much current interest focuses on the potential of such electrodes to serve as links to prostheses capable of restoring lost functions. In 1969 Marks described the growth of frog axons through multiple metal-lined holes drilled through Teflon[®] plates (3). The concept of a regeneration electrode implanted in moving amphibia was proposed by Mannard et al. in 1974 (4). Edell described a chronic neural interface to record from rabbit axons grown through slits in a silicon wafer in 1986 (5). In 1987 Rosen et al. reported the development of an implantable micro-switchboard to decipher and process neural signals (6).

Development of a suitable, long-term neural interface has been limited by several factors. A subnormal and variable degree of nerve regeneration is almost always encountered following injury. Furthermore, functional motor and sensory connections are often lost or mismatched because of inappropriate axonal synapsing. Neural tissue is also prone to scar tissue formation. Therefore, guidance channel/electrode materials should maximize axonal outgrowth and enhance the specificity of axonal connections, while minimizing scar tissue formation.

A prototype neural interface is pictured in figure 1. The device consists of a tubular structure to guide growing axons and a neural interface to serve as a link with peripheral prosthetic devices. The cut nerve ends are fit snugly into the synthetic conduits and are anchored with epineurial sutures. The gap distance between the nerve stumps is controlled by the length of tube used and how far the nerve ends are placed into the tube lumen. Several mammalian animal models have been tested, including mice, rats, rabbits, hamsters, and non-human primates.

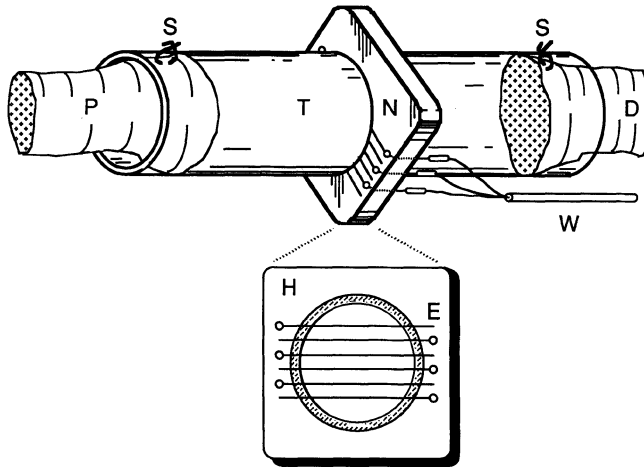


Figure 1. A prototype neural interface with a tubular guidance structure (T) and neural interface (N) which is externalized via insulated wires (W). The proximal (P) and distal (D) nerve stumps are anchored to the tube with microsutures (S). An en face view of the neural interface shows the housing superstructure (H) and the biocompatible electrodes (E).

The regenerated neural structures are evaluated morphologically to quantify the outcome of regeneration for different guidance channel materials tested. Parameters analyzed include: the cross-sectional area of the regenerated nerve cable, the number of myelinated and unmyelinated axons, and the relative percentages of cellular constituents (i.e. epineurium, endoneurium, blood vessels, etc.).

The following properties of synthetic nerve guidance channels appear to be important: transmural permeability, surface charges, surface microgeometry, and growth factor release.

1. Transmural permeability:

The transmural permeability of the guidance channel to watery solutes influences the outcome of nerve regeneration (7). Nerves regenerated through semipermeable tubes display more normal morphological characteristics than nerves regenerated in impermeable or freely permeable tubes (8). Nerve cables regenerated in semipermeable or impermeable tubes are surrounded by an acellular gel and are found free from attachment to the inner wall of the guidance channel (figure 2).



Figure 2. Scanning electron micrograph showing a transverse section through the midpoint of a semipermeable guidance channel after 4 weeks of implantation in a mouse. The nerve cable is separated from the semipermeable inner skin (S) of the channel. Numerous myelinated axon (A) profiles can be seen. The black, circular areas in the cable are blood vessels which have been emptied due to perfusion fixation.

Semipermeable channels with a molecular weight (Mw) cutoff of 50,000 Daltons (D) support regeneration even in the absence of a

distal nerve stump (9). The range of permselectivity is very important since no regeneration is observed if the Mw cutoff is 10^6 D or greater (10). Regeneration does not occur if only the proximal nerve segment is placed into an impermeable tube (9).

These observations suggest that controlled solute exchange between the internal regenerative and external wound-healing environments is essential in controlling regeneration. The process may be modulated by excitatory and inhibitory factors released by the wound-healing environment. Semipermeable channels fitted with neural electrodes show potential for use in amputees since regeneration occurs in the absence of distal neural tissue. A pilot study using blind-ended permselective tubes fitted with 8 - 10 μm thick carbon fibers (to serve as a neural interface) showed that significant numbers of regenerated axons grew from the proximal nerve through the carbon fiber network and toward the distal cap. Regeneration at the level of the dorsal and ventral spinal cord roots also appears possible with the use of semipermeable channels (11). This approach may allow more specific interfacing than with mixed peripheral nerves since ventral roots carry motor information and dorsal roots carry sensory information.

2. Surface charge characteristics

Applied electrical fields and direct DC stimulation influence nerve regeneration. Certain dielectric polymers may be used to study the effect of electrical activity on nerve regeneration in vivo and in vitro. These materials are advantageous in that they provide electrical charges without the need for an external power supply, can be localized anatomically, are biocompatible, and can be formed into a variety of shapes including tubes and sheets.

Electrets are a broad class of dielectric polymers which can be fabricated to display surface charges because of their unique molecular structure. True electret materials, such as polytetrafluoroethylene (PTFE) can be formed so as to exhibit a static surface charge due to the presence of stable, monopolar charges located in the bulk of the polymer (12). The sign of the charge is dependent on the poling conditions. Positive, negative,

or combined charge patterns may be achieved. The magnitude of surface charge density is related to the number and stability of the monopolar charges. Charge stability is related to the temperature at which charging occurs.

Piezoelectric materials such as polyvinylidene fluoride (PVDF) display transient surface charges related to dynamic spatial reorientation of molecular dipoles located in the polymer bulk (13). The amplitude of charge generation is dependent on the degree of physical deformation (i.e. thickness changes) of the polymer structure. The materials show no net charge at rest.

In collaboration with Paolo Dario and others at the University of Pisa, we have fabricated, characterized, and implanted negatively and positively poled PVDF and PTFE tubes as nerve guidance channels. Poled PVDF and PTFE channels contain significantly more myelinated axons than unpoled, but otherwise identical, channels (12, 13). In general, positively poled channels contained larger neural cables with more myelinated axons than negatively poled tubes. It is not currently possible to state whether static or transient charge generation affect regeneration differently.

To test the influence of surface charges on cells and axons growing directly on a synthetic material, strips of poled (i.e. piezoelectric) and unpoled PVDF film were placed lengthwise within the lumen of impermeable silicone elastomer (SE) tubes. Nerves regenerated on nitrocellulose strips placed similarly in an SE chamber formed two bell-shaped structures attached to the nitrocellulose strip. With PVDF material, however, two discrete nerve cables free from attachment to the PVDF strip were observed (figure 3A). The smooth, hydrophobic surface provided by the PVDF may be responsible. Nerves regenerated in close proximity to the poled PVDF strips showed a preferential alignment of cells. Axons and their myelin sheaths as well as blood vessels flattened and elongated parallel to the transverse surface of the piezoelectric PVDF material (figure 3B). Preliminary results with neurons cultured on PVDF and PTFE sheets also suggest an influence of charge generation on neurite extension.

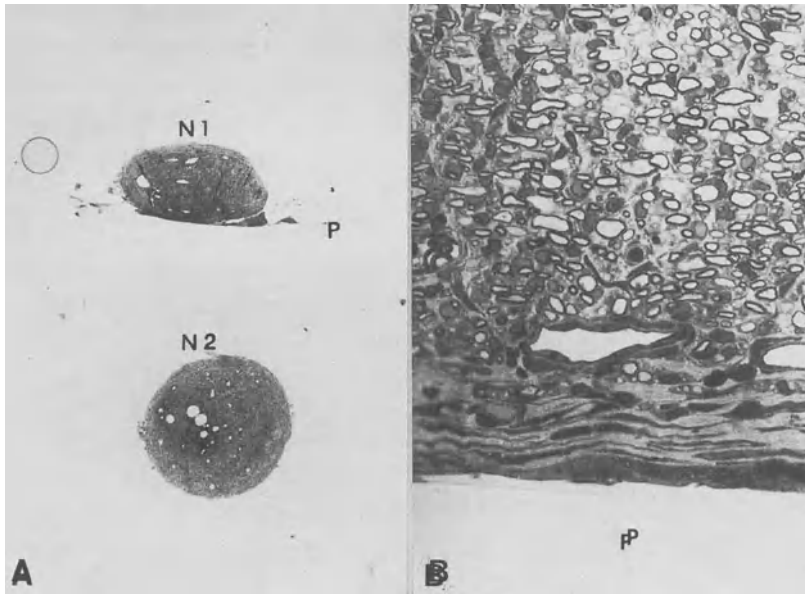


Figure 3. Light micrograph of **A)** a transverse section through the midpoint of a silicone elastomer (SE) tube fitted with a piezoelectric polyvinylidenefluoride (PVDF) strip (P) after 4 weeks of implantation in a rat. The nerve cable (N2) more distant from the PVDF strip is rounder than the nerve cable (N1) growing near the PVDF strip. At higher power **B)**, note that the myelinated axons and blood vessels are elongated in the transverse direction of the PVDF strip in N1. The SE channel was removed in order to process the specimen.

The enhancement of regeneration may be due to electrical influences on protein synthesis, membrane receptor mobility, growth cone motility, cell migration, etc. Proper selection of materials and charge patterns may allow selective growth of two or more nerve cables containing motor or sensory fascicles.

3. Surface microgeometry

The microgeometry of the luminal surface of the guidance channel plays an important role in regulating tissue outgrowth (14). Expanded microfibrillar polytetrafluoroethylene (PTFE) tubes exhibiting different internodal distances (1, 5, and 10 μm) were compared to smooth-walled PTFE tubes. Larger internodal distances result in greater surface irregularity and increased transmural porosity. Rough-walled tubes contained isolated fascicles of nerves dispersed within a loose connective tissue stroma (figure

4A). The greater the surface roughness, the greater the spread of fascicles. In contrast, smooth-walled, impermeable PTFE tubes contained a discrete nerve cable delineated by an epineurium and located within the center of the guidance channel (figure 4B).

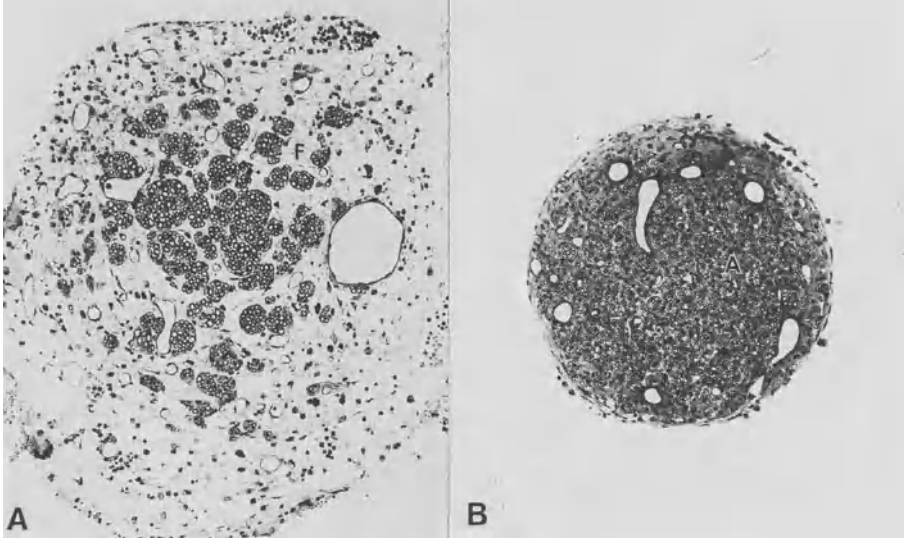


Figure 4. Light micrographs of transverse sections through the midpoint of; **A)** an expanded polytetrafluorethylene (PTFE) channel with a 10 μm internodal distance and a rough inner surface after 12 weeks of implantation in a mouse. Note the axon containing fascicles (F) dispersed throughout a loose connective tissue stroma which fills the lumen of the tube, **B)** an impermeable, smooth-walled PTFE tube after 12 weeks of implantation in a mouse. Note the nerve cable containing myelinated axons (A), ensheathed by an epineurium, and separated from the PTFE tube. The PTFE channels were removed to process the specimen.

These studies suggest that the microgeometry of the guidance channel lumen modulates the nerve regeneration process. Wall structure changes may alter the protein and cellular constituents of the regenerating tissue bridge. Further information regarding the control of fascicle formation and location may aid in the quest for an effective neural interface. The physical isolation of fascicles containing motor or sensory information from specific muscle or end organ structures would simplify signal processing and analysis. In addition, the ability to direct

fascicle patterns would minimize the amount of microcircuitry needed to decode and transcribe the signals.

4. Growth factors release

The release of growth factors and other substrates from synthetic guidance channels may improve the degree and specificity of neural outgrowth. Channels releasing basic fibroblast growth factor (b-FGF) increase the distance over which axons will regenerate (15). Channels composed of an ethylene-vinyl acetate copolymer can be fabricated to release incorporated macromolecules in a predictable manner. The geometry of the drug releasing structure affects the drug-release kinetics. It is possible to restrict drug-release to the luminal side of the guidance channel by coating its outer wall with a film of pure polymer.

The release of growth factors which preferentially control the survival and outgrowth of specific motor and sensory neuronal subpopulations may enhance the sensitivity of neural interfaces. Nerve growth factor (NGF) and b-FGF control sensory neuronal survival and outgrowth and other putative growth factors are thought to control motor neuronal survival and outgrowth. Growth factors incorporated within a multicompartiment guidance structure housing neural electrodes may provide more accurate motor/sensory signal processing. Growth factors released by guidance channels may also allow regeneration over large nerve deficits, an important consideration in nerve injuries with severe tissue loss.

Discussion

In peripheral nervous system injuries, much of the afferent/efferent signal transduction and central processing system (i.e. brain and spinal cord) remains intact. The technology of motor and sensory prostheses has advanced to the point where the reproduction of normal function is feasible. Interfacing prosthetic devices with surviving efferent and

afferent axons could restore motor function and sensory feedback. The development of an appropriate neuroelectrical connection has proven difficult. Much work has focused on the use of biomaterials in elucidating the mechanisms underlying peripheral nervous system regeneration. In the case of neural interfaces, biomaterials serve multiple functions. They must: 1) insure optimal regenerating conditions to maximize axonal outgrowth and function, 2) provide a microenvironment suitable for chronic interfacing with sensitive neural tissue, and 3) provide an interface which allows for the accurate extraction and input of signals to and from motor and sensory neurons with various target structures.

Nerve regeneration following injury is highly variable and rarely reaches normal pre-injury levels. New biological and technical factors, including the use of biomaterials, may aid in overcoming this problem. Neural interfaces may be less hampered by these limitations since microelectronics and prosthetic devices will serve to augment or replace normal function. Interfacing with only several axons originating from the same fascicle may be sufficient to provide adequate function. The permeability, electrical, physical, and drug-release parameters of guidance channels may be tailored and used in combination to optimize regeneration.

The second major problem, that of scar tissue formation and neuronal damage in chronic implants, has been a limiting factor in the development of useful long-term electrode interfaces. We believe that the proper choice and/or development of new biomaterials is essential toward this aim.

The problems related to target specificity may be the most challenging. The combined use of biomaterials and growth factors may address this issue. Surface microgeometry can be controlled so as to isolate fascicles, electrical charges can be used to guide axons or to separate nerve cables, and growth factors can be used to guide and specify outgrowth, even over long distances. Another interesting possibility is offered by the direct regeneration electrode repair of ventral and dorsal spinal cord roots. This model eliminates some specificity problems and may be

useful in developing some devices. The problem of limited dorsal root regeneration into the spinal cord must, however, be addressed.

In conclusion, biomaterials provide new avenues in the understanding and development of regeneration electrode devices. By maximizing neural regeneration, specifying outgrowth patterns, and providing a biocompatible electrode network, it may be possible to design implantable devices capable of chronic neural signal transduction in humans.

ACKNOWLEDGEMENTS

This work was partially supported by a grant from the Whitaker Foundation and NIH grant NS-26159 to P.A.

REFERENCES:

1. Cajal S.R. Degeneration and regeneration of the nervous system. Oxford University Press, London, 1928.
2. Aebischer P. The role of biomaterials in peripheral nerve regeneration. "Tissue Engineering", UCLA symposia on Molecular and Cellular Biology, New Series; Eds. R. Skalak and C.F. Fox, Alan R. Liss, Inc, N.Y., N.Y.; 104; 257-262, 1988.
3. Marks A.F. Bullfrog regeneration into porous implants. Anat. Rec. 163; 226, 1969.
4. Mannard A., Stein R.B., Charles D. Regeneration electrode units: Implants for recording from single peripheral nerve fibers in freely moving animals. Science 183; 547-549, 1975.
5. Edell D.J. A peripheral nerve information transducer for amputees: Long-term multichannel recordings from rabbit peripheral nerves. IEEE Trans. Bio. Eng. 33; 203-214, 1986.
6. Rosen J.M., Grosser M. Nerve repair at the axon level: A merger of microsurgery and microelectronics. In: Artificial Organs; Ed. J.D.Andrade, VCH Publishers Inc, N.Y., N.Y.; pp 583-594, 1987.
7. Aebischer P., Valentini R.F., Winn S.R., Kunz S.K., Galletti P.M. Are guidance channel composition and structure important factors in peripheral nerve regeneration? Soc. Neurosci. Abstr. 12; 699, 1986.
8. Valentini R.F., Aebischer P., Winn S.R., Galletti P.M. Collagen- and laminin-containing gels impede peripheral nerve regeneration through semipermeable nerve guidance channels. Exp. Neur. 98; 350-356, 1987.
9. Aebischer P., Guénard V., Winn S.R., Valentini R.F., Galletti P.M. Blind-ended semi-permeable guidance channels support peripheral nerve regeneration in the absence of a distal nerve stump. Brain Res. 454; 179-187, 1988.
10. Aebischer P., Guénard V., Brace S. Peripheral nerve regeneration through blind-ended semipermeable guidance channels: effect of the molecular weight cut-off. J. Neurosci. 9; 3590-3595, 1989.

11. McCormack M., Guénard V., Goddard M., Beauregard A., Brace S., Aebischer, P. Regeneration of adult dorsal and ventral root axons in semipermeable guidance channels. Soc. Neurosci. Abstr. 15; 1257, 1989.
12. Valentini R.F., Sabatini A.M., Dario .P, Aebischer P. Polymer electret guidance channels enhance peripheral nerve regeneration in mice. Brain Res. 480; 300-304, 1989.
13. Aebischer P., Valentini R.F., Dario P., Domenici C., Galletti P.M. Piezoelectric guidance channels enhance regeneration in the mouse sciatic nerve after axotomy. Brain Res. 436; 165-168, 1987.
14. Guénard V., Aebischer P. Effect of guidance channel surface microgeometry on peripheral nerve regeneration. Soc. Neurosci. Abstr. 15; 317, 1989.
15. Aebischer P., Salessiotis A.N., Winn S.R. Basic fibroblast growth factor released from synthetic nerve guidance channels is a potent mediator of nerve regeneration in vivo. J. Neurosci. Res. 23; 282-289, 1989.

Regeneration-Type Peripheral Nerve Interfaces for Direct Man/Machine Communication

Gregory T. A. Kovacs and Joseph M. Rosen

Stanford University Department of Electrical Engineering, Stanford University Department of Surgery (Division of Plastic and Reconstructive Surgery) and the Department of Veterans Affairs Hospital, Palo Alto

Address: G. T. A. Kovacs, Stanford University, CIS 130, M/C 4070, Stanford, CA, 94305, U.S.A.

Abstract

The goal of all man/machine interfaces is to connect the human nervous system, directly or indirectly, to machines. An overview of man/machine interfaces is presented, discussing the various techniques presently available. The development of a promising type of direct interface to peripheral nerves is presented, from the origins of the idea to the work in progress. The basic idea of these regeneration-type neural interfaces is that a planar array of microelectrodes, fabricated on a substrate that is perforated by a plurality of *via holes*, is placed between the ends of a surgically severed peripheral nerve. Axons regenerate through the *via holes* and become spatially fixed with respect to the microelectrodes, providing a stable interface. If silicon is used as the substrate for the array, integrated microelectronic circuits can be used to control communication between the nerve and external circuitry. The goal of this project is to develop such a device to interconnect the nerves in an amputation stump and a robotic motor/sensory limb prosthesis.

Introduction

High-performance man/machine interfaces have long been sought to efficiently couple the human nervous system to machines. One such device, which promises access to a large portion of the information bandwidth of the nervous system, is the regeneration-type peripheral nerve interface. The basic concept of these neural interfaces is that a planar array of microelectrodes, fabricated on a substrate that is perforated by a plurality of holes (referred to as *via holes*), is placed between the surgically severed ends of a peripheral nerve. Axons regenerate through the *via holes* and become spatially fixed with respect to the microelectrodes, providing a stable interface. If silicon is used as the substrate for the array, integrated microelectronic circuits can be used to control communication between the nerve and circuitry. This should eventually permit permanent access to neural signals at or near the level of individual axons.

The article begins with a general overview of man/machine interfaces, emphasizing the relative strengths and weaknesses of a variety of methods. This is followed by a review of the history and state-of-the-art in regeneration-type electrode arrays. The development and testing

of the basic technologies to fabricate the desired device is discussed. A proposed use of the neural interface as a link to a motor/sensory limb prosthesis is then presented, followed by comments on additional uses of such interfaces.

Overview of Man/Machine Interfaces

The primary objective of the research described herein is to progress toward the development of a *direct* man/machine interface for clinical use. The continuum of hybrid man/machine systems, encompassing the range between human and machine, is illustrated in Figure 1 below. At present, interface with the central nervous system (CNS) or peripheral nervous system (PNS) is commonly accomplished via several *indirect* techniques. The most common man/machine interfaces for *input* to the machine are mechanical devices such as keyboards, switches, dials, touch-screens, and “mice.” Other approaches, including speech recognition, eye tracking, and hand gesture recognition with devices such as the *data glove*¹, are being developed to provide increased information bandwidth and a more natural interface. The purpose of these devices is to translate users’ desires into the appropriate actions of the machines with which they are communicating. Typical machine *output* devices are visual displays, hard-copy devices such as printers, or sometimes speech synthesizers for purely language-based information. Here, the objective is to convey as much information as possible from the machine to the user. Other forms of information, such as tactile and other sensory modalities, are seldom used but have important implications for the prosthesis applications discussed below. An interesting overview of several types of novel man/machine interfaces can be found in Stewart Brand’s, “The Media Lab” [1].

Among present interfaces, the concentration has naturally been on the highest bandwidth, easiest to use, *externally* available input/output channels. For *input to* machines, mechanical (hands) or, occasionally, speech pathways are used; and for *output from* machines, visual and auditory sensory channels are chosen. Direct access to the nervous system (if it can retain a large fraction of the potentially available information bandwidth) should provide sizeable increases in performance over these types of interfaces and allow for replacement or bypass of defective nervous tissue.

Indirect methods of accessing the information in the nervous system include surface recording techniques, such as *electromyography* (EMG) or *electroencephalography* (EEG), which provide ensemble averages of the neural action potentials occurring therein. While EMG-based techniques are commonly used to control motor prostheses [2], and EEG’s have been used to input information to machines [3], [4], each technique makes use of only a minute fraction of the underlying neural information bandwidth and both are clearly unidirectional.

¹The data glove is a device worn on the human hand to measure joint angles and hand position in real time (manufactured by VPL Inc., Redwood City, CA).

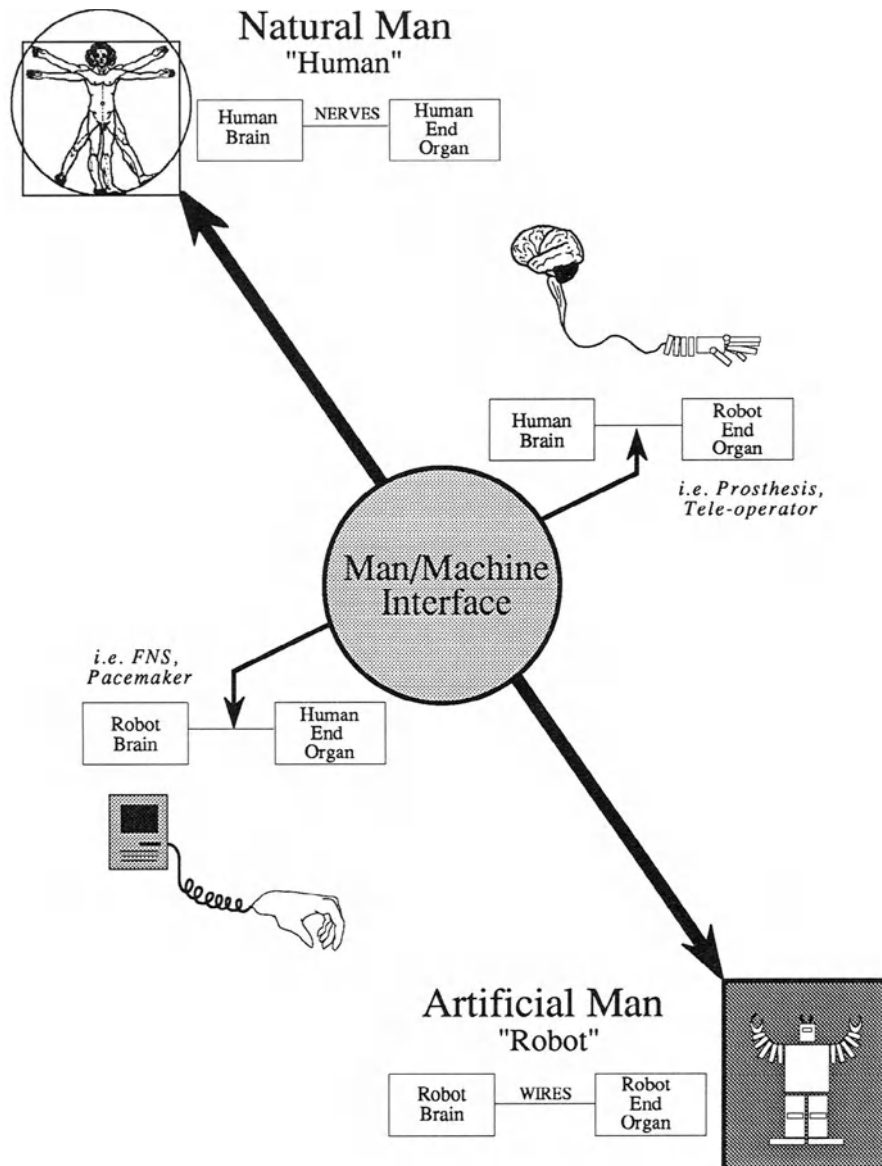


Figure 1: Illustration depicting the continuum between natural man, or "human," and artificial man, or "robot." It is the region between these two extremes that constitutes the domain of the man/machine interface. In all cases, a controller, or "brain," is connected by some means to an effector or sensor ("end organ"). The control of a human end organ by a machine is fairly common in medical applications such as cardiac pacemakers, and is being used for computer-controlled walking via *functional neuromuscular stimulation*. The control of a robot end organ by the human brain is common in applications where the interface is *external* to the body, such as the operation of machinery with manual controls. Direct, neural control of a robot end organ, such as a limb prosthesis, is the goal of this research and has yet to be achieved. (Drawing by Kathryn M. Clark.)

Surface stimulation of the nervous system has been used to stimulate muscles for computer-controlled arm and hand use [5], for ambulation [6], and for exercise in paralyzed individuals [7]. These transcutaneous *functional neuromuscular stimulation* (FNS) devices meet with considerable success despite the large currents that must be applied to the skin to obtain the desired muscular function. The clinical use of *transcutaneous electrical nerve stimulation* (TENS) devices for pain control, which apply lower stimulation currents (usually below the threshold of perception) through transcutaneous electrodes, is relatively common [8]. Newer techniques, such as incorporating external electrodes in garments, have eliminated much of the inconvenience of placing individual electrodes on the skin for both transcutaneous FNS and TENS devices [9].

Stimulation systems that use implanted electrodes (or are implanted in their entirety) are common in some applications, such as cardiac pacemakers, but have yet to make clinical inroads into the area of sensory and motor interfaces. Intramuscular and *epimysial* (attached to the muscle belly) electrodes have been in use for several years to electronically control human hand and arm movements [10], [11], and ambulation [12]. These devices, while using considerably less stimulation current than comparable transcutaneous FNS systems, stimulate muscle tissue in an “unnatural” fashion, failing to make use of the normal neuromuscular connections of the peripheral nerves.

Several direct sensory prostheses have demonstrated remarkable success, such as the cochlear auditory prostheses developed at Stanford University, the University of California, San Francisco (UCSF), the University of Melbourne, Australia, and other groups (reviewed in White [13] and Leake, et al. [14]). One of the long-term goals for sensory prostheses has been the development of usable visual prostheses. Despite efforts such as that of Brindley [15] to stimulate the visual cortex of a human volunteer (in whom an array of 80 metal electrodes in a flexible plastic carrier was surgically implanted over the visual cortex), this has remained an elusive goal. Nonetheless, these projects have resulted in several demonstrations of direct stimulation of CNS tissue using implanted electrode arrays. The National Institute of Neurological Disorders and Stroke (NINDS) of the National Institutes of Health is currently funding the ongoing development of directly implantable, silicon-based, CNS interface devices that contain on-chip signal-processing electronics, as discussed in Anderson, et al. [16], and Najafi, et al. [17].

Direct FNS using extraneural electrode “cuffs” over entire nerves has been demonstrated and is reviewed by Naples, et al. [18]. With such devices, muscles are stimulated in a more natural fashion (by the nerves themselves), but the resolution of the control over muscle function is limited because the stimulus currents are applied grossly to the nerve. In addition, as with all FNS systems to date, the normal feedback information from the muscles, tendons, joints, and skin is not available to the control algorithm. External measurements of joint angles are generally used as inputs to closed-loop control algorithms, although the use of natural tactile receptors as feedback sources has been proposed by Hoffer, et al. [19].

A limitation of the interfaces mentioned above is that they are all *unidirectional*, while nearly every part of the nervous system carries bidirectional signals. Clearly, bidirectional interfaces will be required if “complete” replacements for limbs are ever to be realized. To make use of the massive information bandwidth of peripheral nerves, such a device would ideally interface at the level of single or small groups of *axons*, the individual nerve fibers that carry information in the PNS. This information is encoded in the temporal firing patterns of individual axons (by pulse-frequency modulation) and the global patterns of firing in a peripheral nerve.

Potentially bidirectional interfaces, which are currently used only in laboratory situations, include extracellular microelectrodes and intracellular needle microelectrodes (which must be positioned, using micromanipulators, so that their tips penetrate the cell membranes of individual axons). Intracellular electrodes cannot be utilized for practical prostheses since it is not possible to implant them chronically or in large enough numbers to access a clinically usable amount of information.

The use of small extracellular electrodes is generally limited by their lack of spatial stabilization with respect to the nerves in which they are implanted. A major advantage of extracellular microelectrodes is that they can be arranged in arrays on a common substrate, permitting simultaneous access to multiple groups of axons. Thin-film fabrication techniques, first applied to extracellular microelectrode arrays by Wise, et al. [20], [21], made mass production of such devices practical. Since the fabrication of the first such devices in the early 1970's, several groups have fabricated and used them *in vitro* and *in vivo*² (reviewed in [22]).

Regardless of the selectivity of microelectrodes³, stable functional mappings between axons and microelectrodes cannot be made unless they are positionally fixed with respect to each other. As previously mentioned, the basic idea of regeneration-type neural interfaces is to integrate the microelectrode array into the neural tissue by allowing the axons to grow through a plurality of via holes through the substrate of the array. Thus, as discussed in detail below, the regenerative capacity of the PNS is used to spatially fix the locations of the microelectrodes with respect to the axonal population with which they communicate.

History of Regeneration-Type Neural Interfaces

It is well known that peripheral nerves of vertebrates (including humans) will regenerate if severed and re-apposed or crushed. Axons distal to the injury site degenerate in a process referred to as *Wallerian degeneration*. Schwann cells in the distal portion of the nerve phagocytize the degenerating axons but remain aligned in “empty” tubes (sometimes referred to as “Bands

²*In vivo* experiments have been limited, in general, to acute preparations.

³Selectivity herein refers to the spatial resolution of an electrode in terms of the number of axons with which it interacts electrically.

of Büngner”) into which regenerating axons can grow [23]. If the proximal portion of the nerve is close enough to the distal portion for regeneration to occur, *growth cones* of the regenerating axons will find their way down these tubes. If sensory end organs or motor end-plates are present at the end of the tubes, new terminal boutons will form, completing the regeneration process. Of course, the organization of the nerve is then “scrambled” since each regenerating axon has only a very low probability of re-connecting to the correct sensory end organ, or muscle fiber, distally. This problem manifests itself clinically in nerve injury cases as a return of only coarse or “protective” sensibility in the affected limbs.⁴ On this basis, surgical repair of severed peripheral nerves is commonly carried out today [23].

The basic concept of fabricating regeneration-type neural interfaces for use in the PNS is not new, but devices with on-chip microcircuits have yet to be realized. One of the main reasons for this has been a lack of development of suitably durable and biocompatible fabrication techniques that are also compatible with the inclusion of active microcircuits. The history of work toward this goal is outlined below.

In 1969, Marks [24] carried out experiments to, in her own words, “test the possibility that cut nerve fibers will regenerate into a chronic multiple microelectrode consisting of insulating material with small metal-lined holes having individual leadouts.” She demonstrated that bullfrog sciatic nerve axons would regenerate through porous implants placed at the point of transection of the nerve. Regeneration of sciatic and optic nerve axons⁵ through pieces of Teflon sponge, “continuous pore” Teflon, and arrays of 25 μm (i.d.) gold cylinders embedded in Teflon was observed. Unfortunately, the goal of the project, chronic connections to the optic nerve, met with little success [25].

The work of Marks, and that which followed, was motivated by the idea that regenerating axons could be confined within insulating holes through a substrate on which microelectrodes had been fabricated. (These pairs of holes and electrodes would form so-called “regeneration electrodes.”) Thus, the microelectrodes would be spatially fixed with respect to the axons. As explained above, a fixed relationship between the neural tissue and the microelectrodes is required to obtain a high-resolution interface to the information carried in peripheral nerves. Naturally, many such microelectrodes would be required since information in the nervous system is carried in a parallel, distributed manner.

In 1973, Llinás, Nicholson, and Johnson [26] published a remarkable paper describing how a silicon-based microelectronic neural interface might someday be implemented. They proposed a prototype array of regeneration electrodes that could be placed between the severed ends of a peripheral nerve, as shown in Figure 2 below. The device would have circular gold electrodes through which via holes would be laser drilled. Each electrode would be connected, via multiplexing circuitry, to output amplifiers for recording (as shown in Figure 3 below).

⁴If the nerve leads to a “dead-end,” such as those found in amputation stumps, distal connections would be inconsequential. Interfacing to such nerves is the objective of this project.

⁵Regeneration of the optic nerve is not normally possible in mammals.

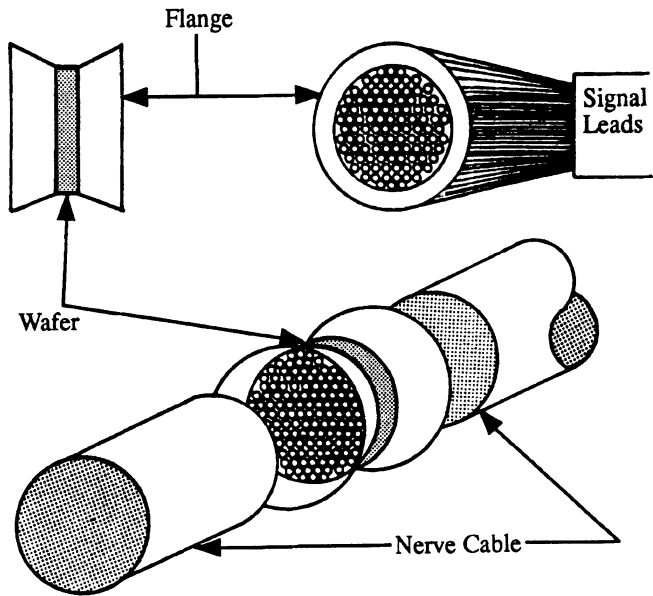


Figure 2: Diagram illustrating the surgical coupling of a silicon-substrate neural interface device between the severed ends of a peripheral nerve, as proposed by Llinás, et al. [26]. (Used with permission.)

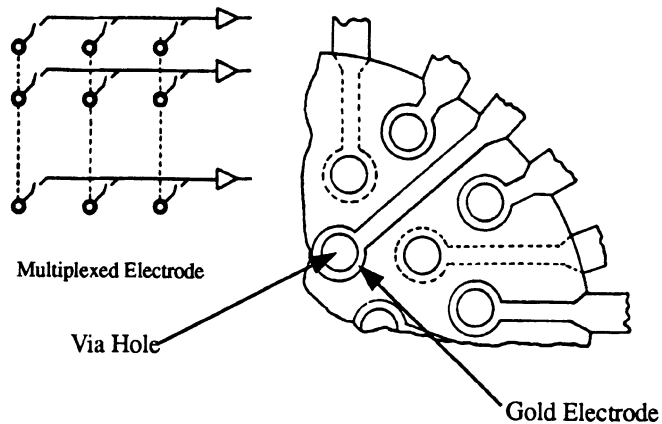


Figure 3: Diagram illustrating the laser-drilled microelectrodes and multiplexing circuit configuration proposed by Llinás, et al. [26]. (Used with permission.)

Llinás, et al., also proposed using on-chip microcircuits to address individual microelectrodes in the array of via holes (each containing a regenerated peripheral nerve axon) and to communicate the recorded signals to external processing electronics. Such on-chip multiplexing schemes are necessary due to the topological limitations imposed by tightly-spaced arrays of large numbers of microelectrodes. With a reasonable number of electrical interconnect layers (one to three in most integrated circuit fabrication processes), it is impossible to directly connect to all electrodes in a two-dimensional array without multiplexing. In addition, the requirement to minimize the number of off-chip connections makes this the only practical approach. Figure 4 below shows the overall architecture of the neural interface system proposed by Llinás, et al. Few of the details of realizing such a design were worked out within the scope of their paper, and no devices were designed or fabricated. However, the basic idea for a high-density, multiplexed, regeneration-type neural interface had been proposed and influenced many later researchers.

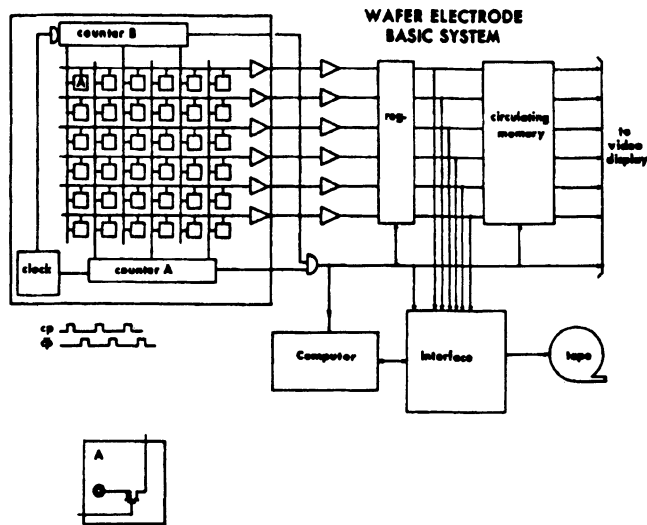


Figure 4: The overall architecture of the neural interface proposed by Llinás, et al. [26]. (Used with permission.)

In the following years, several groups studied regeneration electrodes made from non-semiconductor materials. The first recordings from a “tubular regeneration electrode array” appear to be those obtained by Mannard, Stein, and Charles in 1974, at the University of Alberta, Canada [27]. They fabricated an array of ten electrodes (as illustrated in Figure 5 below) by mechanically drilling 100 μm diameter holes into a 700 μm thick epoxy recording unit containing 77 μm diameter Teflon-coated silver wires. A polyethylene cuff was used to couple the device to the severed nerve.

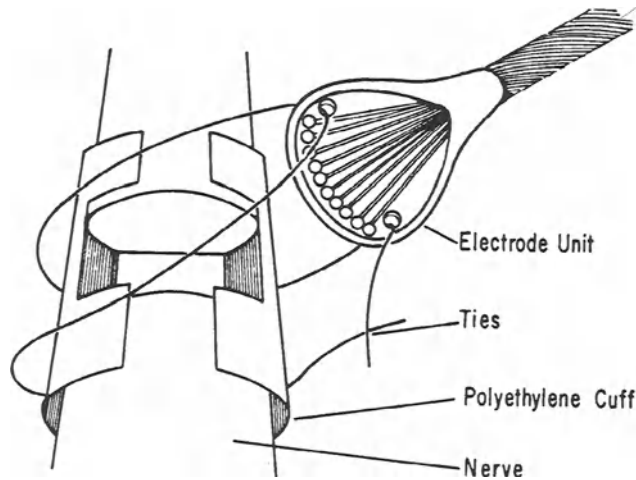


Figure 5: Drawing of the tubular regeneration electrode of Mannard, et al. [27]. The device consisted of 100 μm holes mechanically drilled into an epoxy frame containing 77 μm diameter Teflon-coated silver wires. (Used with permission.)

Afferent (sensory) and efferent (motor) action potential recordings were obtained and neurostimulation demonstrated using such arrays implanted in *Xenopus laevis*. Each hole was observed to contain between one and twenty-nine regenerated axons. Although transmission electron microscope (TEM) and conduction velocity studies revealed less than normal myelination of the axons, control nerves allowed to regenerate without obstruction by the recording array showed a similar lack of re-myelination.

A two-dimensional “tubular regeneration array” comprised of stacked planar arrays of conductive traces on flexible substrates, was described by Loeb, Marks, and Beatty in 1977 [28]. The arrays consisted of 10–15 μm square holes, 300–1200 μm long, on 25 μm centers, as illustrated in Figure 6 below. While the arrays were apparently not successfully used *in vivo*, these researchers derived theoretical models for signal amplitudes expected from micro-electrodes located within the holes and expressed their concern that oxygen diffusion into the holes may be a limiting factor to regeneration of viable axons.

In 1978, Matsuo, Yamaguchi, and Esashi [30] reported the fabrication of recording arrays of $200 \times 200 \mu\text{m}$ square via holes etched using a pyrocatechol wet-etching technique (this micromachining technique is described in [29]) in a $\langle 100 \rangle$ orientation silicon wafer of approximately 200 μm in thickness. They included novel MOSFET (metal oxide semiconductor field-effect transistor) buffer transistors *in* the via holes utilizing a unique “tubular” architecture, shown in Figure 7 below. The gates of the tubular MOSFETs lay within and encircled

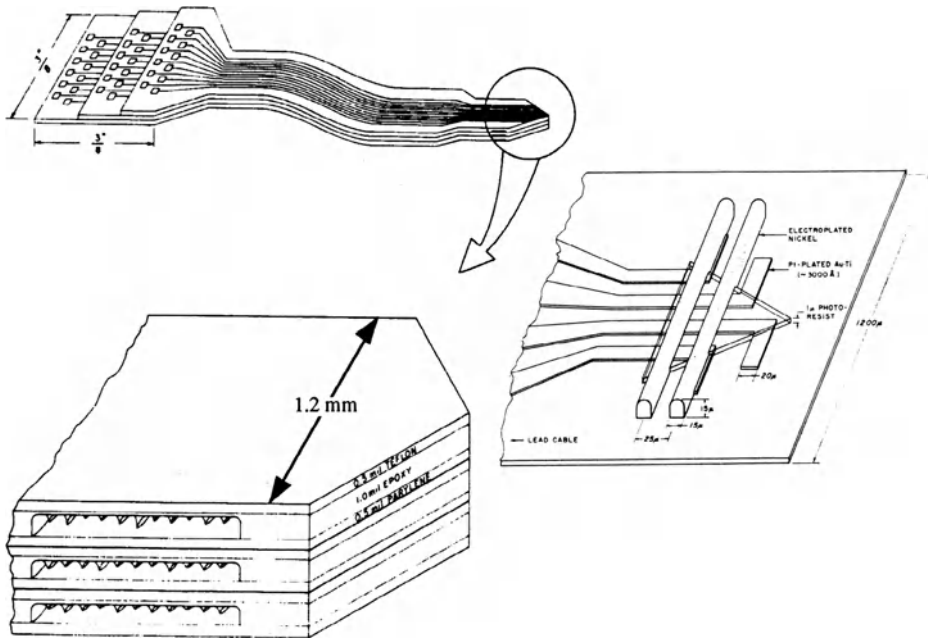


Figure 6: Diagram of the “tubular regeneration array” proposed by Loeb, Marks, and Beatty in 1977 [28]. Stacked planar arrays with etched tubular openings were intended to be placed in the path of regenerating axons. (Used with permission.)

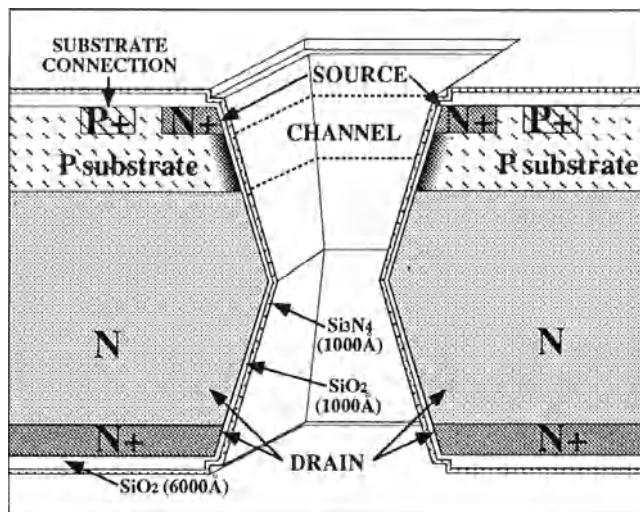


Figure 7: Illustration of the regeneration-type recording structure of Matsuo, et al. [30]. The unique, “tubular” MOSFET design used by this group is depicted above, showing that the “gate” was inside the $200 \times 200 \mu\text{m}$ via hole, with the source and drain on opposite sides of the silicon die. (Drawing not to scale.) [Drawing by G. Kovacs, after Matsuo, et al.]

the via holes, with their sources and drains on opposite sides of the wafer. Ten such transistors were fabricated on each device, which was approximately $4\text{ mm} \times 4\text{ mm}$ in size. Apparently, no recordings were made due to low signal-to-noise ratios that were attributed to high thermal noise levels⁶ [31].

The first recordings obtained with a silicon substrate regeneration array were reported in the doctoral dissertation of Edell, published in 1980 [32]. Edell considered, but did not pursue, the approach of laser-drilling large arrays of via holes. Instead, using potassium hydroxide etching of <110> silicon wafers (this technique is described in [29]), Edell etched large ($120\text{ }\mu\text{m} \times \approx 1.5\text{ mm}$) slots in a substrate with final thickness of $140\text{ }\mu\text{m}$. Platinum/tantalum interconnect metallization was applied and passivated with chemical vapor deposited silicon nitride and a coating of silicon dioxide. Contacts in the passivation layer were selectively opened to expose ten $650\text{ }\mu\text{m}^2$ microelectrode sites, where gold was deposited.

A scanning electron microscope (SEM) view of one of the completed devices is shown in Figure 8 below. Teflon-coated silver wires were ultrasonically bonded to the bond pads and insulated with multiple epoxy coats. A final multi-layer silastic casing was then applied, serving as the surgical coupler to hold the severed ends of the nerve against the device. Recordings with amplitudes on the order of $150\text{ }\mu\text{V}$ P-P were obtained from such devices implanted in the sciatic nerves of rabbits [32], [33] and represented the action potentials from fairly large numbers of axons in proximity to each microelectrode. Edell not only reiterated the idea of using smaller laser-drilled via holes for axonal regeneration, but also used a silastic-based surgical coupler whose basic design would be modified and used in research at Stanford University.

To determine if individual or small groups of regenerating axons could be isolated in small via holes, laser drilling of such holes was undertaken by Rosen, et al. [34], as initially proposed (but not implemented) by Llinás, et al. [26], and Edell [32]. These preliminary “blank” (without microelectrodes or microelectronic circuits) devices were made from N-type <100> orientation silicon wafers. The wafers were lapped, polished, and diced (following laser drilling) to dimensions of $1\text{ mm} \times 1\text{ mm} \times 100\text{ }\mu\text{m}$ in thickness. Arrays of 50×50 via holes with entrance diameters of $50\text{ }\mu\text{m}$ and exit diameters of $25\text{ }\mu\text{m}$ (shown in Figure 11 below) were initially fabricated⁷ with a CO_2 laser mounted on a computer-controlled X-Y stage. Subsequent refinements in positioning and the use of a YAG (*yttrium aluminum garnet*) laser⁸, made it possible to drill more cylindrically-shaped holes with entrance and exit diameters of $\approx 8\text{ }\mu\text{m}$, approximating the average diameter of a typical peripheral nerve axon.

⁶The high thermal noise levels, presumably resulting from the unusual channel structure of the “tubular” MOSFETs, was caused by high drain-to-source resistance (R_{ds}) values.

⁷The devices were laser-drilled by Dr. T. R. Anthony, General Electric Co., Schenectady, NY.

⁸The devices were laser-drilled by Dr. J. B. Gaylord, P.M. Industries Inc., Portland, OR.

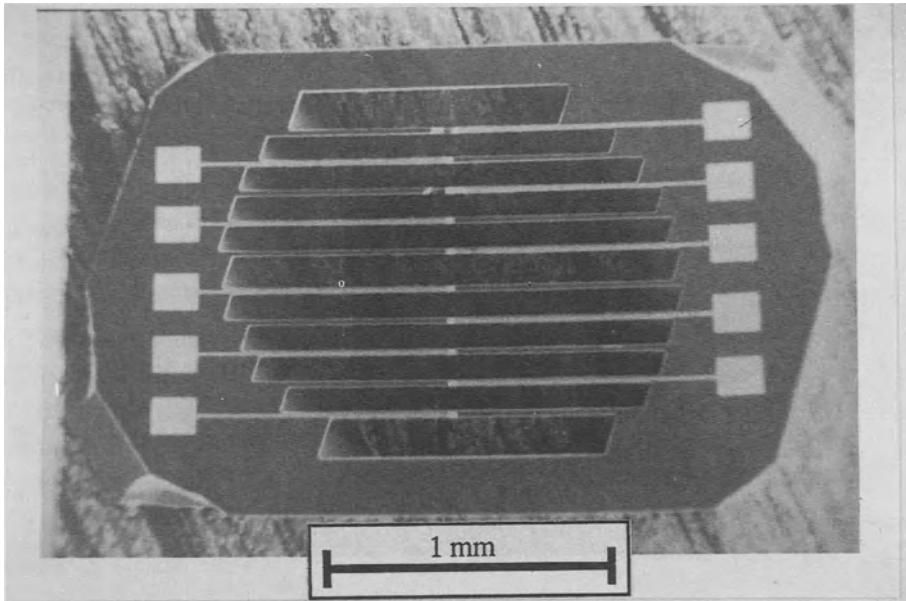


Figure 8: SEM view of regeneration-type recording microelectrode array designed by Edell [32]. The gold microelectrode sites can be seen as the small bright squares in the center of the image, between the slots in the silicon. (magnification = 48X) [Photograph courtesy of Dr. D. J. Edell.]

The blank silicon chips were mounted in polycarbonate or resorbable GTMC (glycolide trimethylene carbonate) surgical couplers⁹ that were micro-cast, as opposed to being molded *onto* the silicon device as done by Edell [32]. The chip/coupler assemblies were implanted between the surgically severed ends of rat and primate nerves. An illustration of one of the implanted assemblies is shown in Figure 9 below.

After allowing time for regeneration of the axons, histological and electrophysiological studies were carried out. This work verified that individual or small groups of physiologically functional axons could regenerate through such small via holes. Figure 10 below is an SEM view of monkey axons that had regenerated through 8 μm diameter via holes in a laser-drilled, blank silicon substrate, biopsied six months post-operatively.

⁹The surgical couplers were manufactured by Davis and Geck, Inc., Pearl River, NY.

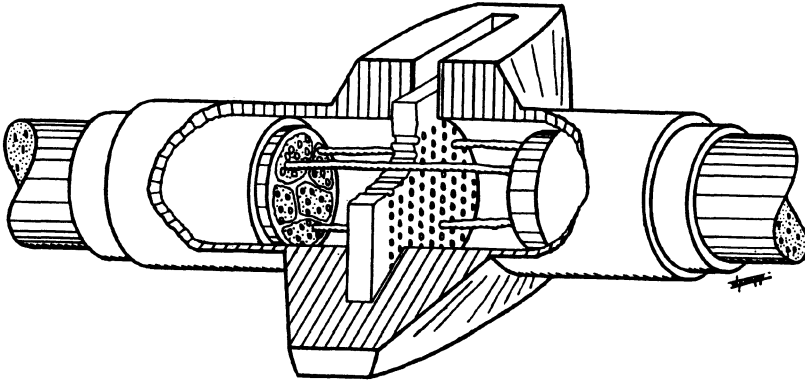


Figure 9: Drawing showing re-apposed ends of a peripheral nerve held against a blank silicon device perforated by via holes, in a surgical coupler as used by Rosen, et al. [34]. Axons that have regenerated through the via holes in the silicon are shown in the cut-away view. (Drawing not to scale.) [Drawing by W. R. Knapp.]

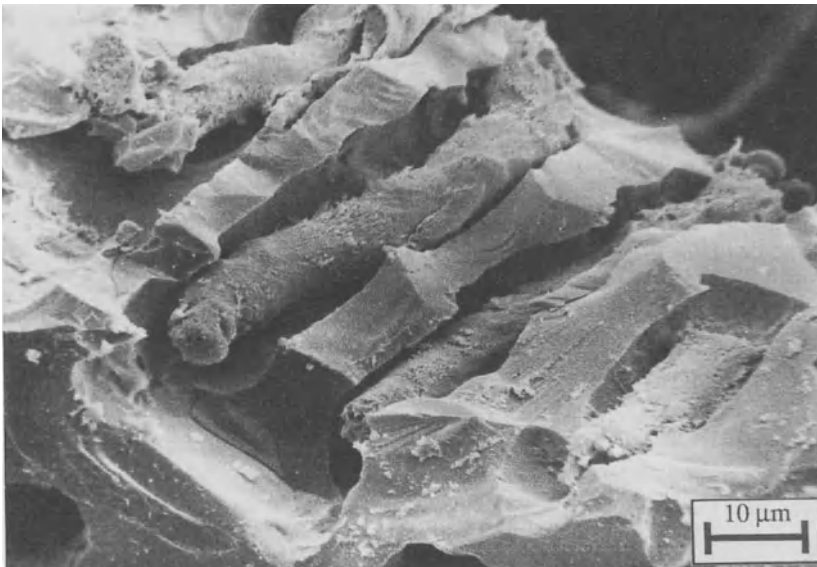


Figure 10: SEM view of monkey axons that had regenerated through via holes in a laser-drilled blank silicon substrate, biopsied six months post-operatively. (magnification = 1512X)

Unfortunately, the laser-drilling process is not compatible with the co-fabrication of active microelectronic circuits on the same silicon substrate, due to damage to the surface and substrate [35] and to beam positioning errors [36]. Figure 11 below shows the visible damage

to the silicon caused by the laser-drilling process. Positioning errors presently associated with laser-drilled via holes would prevent the precise alignment required to any underlying microcircuits. In addition, available X-Y stages for laser drilling limit drilling rates to ≈ 2 holes per second [35], or over 20 minutes per device with a 50×50 array of via holes. For these reasons, the development of a plasma-etching process for via hole fabrication, compatible with Co-fabrication of integrated microelectronics, was undertaken (as discussed below).

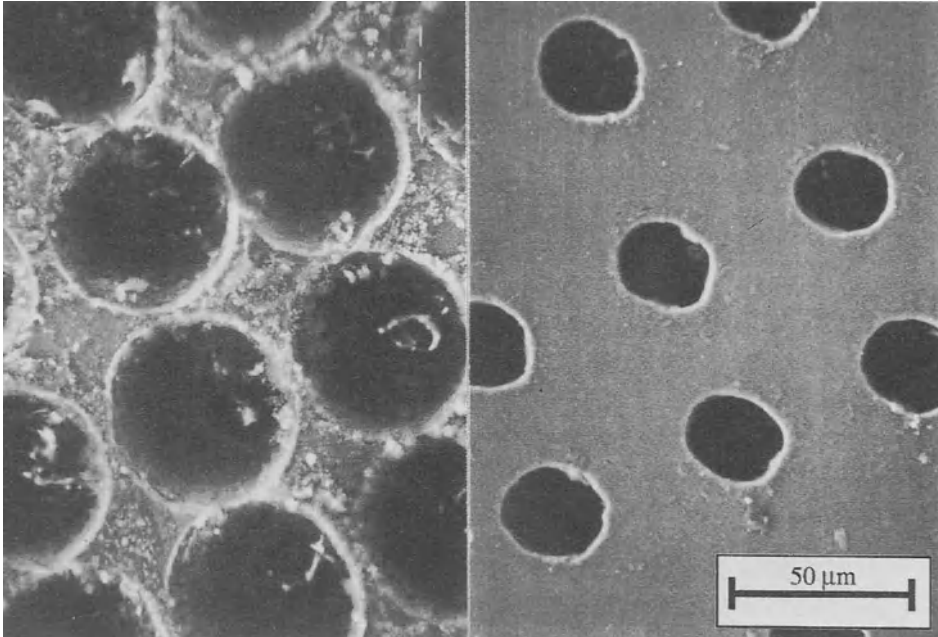


Figure 11: SEM views of the front and back sides (left and right, respectively) of early laser-drilled via holes in a blank implant device, showing surface damage to the silicon. (magnification = 600 X)

The Stanford/DVA Neural Interface

The goal of this research is to deliver a clinically useful prosthetic device to amputees, and the majority of the funding for the project has been provided by the United States Department of Veterans Affairs for this purpose. In order to meet the requirement for a flexible, chronic man/machine interface, a regeneration-type neural interface capable of consistently sensing *and* stimulating action potentials in individual or small groups of peripheral nerve axons is under development.

The present approach is to provide an individual neural interface for each *fascicle* (an internal sub-unit of a peripheral nerve, containing a functionally-related group of up to several thousand axons) of a major nerve. The size of the microelectrode/via hole array for such a

monofascicular interface is constrained to approximately a 1 mm² surface area, corresponding to the cross-sectional area of the fascicle. The microelectrodes are arranged in a two-dimensional grid, at densities to be determined by their selectivity, but potentially approaching those of the axons in peripheral nerves (1,000–2,000 axons per mm²) to maximize access to the information present. The intimate contact of the microelectrodes with the axons, coupled with the anisotropic conductivity of peripheral nerves, should provide for good signal selectivity between microelectrodes.

Development Of Fabrication Techniques

Every effort has been made to implement a technology that can eventually be commercialized and used to produce devices at a reasonable cost. Care was taken to use only processing techniques compatible with commercial CMOS (Complementary Metal-Oxide Semiconductor, a low-power circuit technology) integrated circuit fabrication so that the final arrays could be produced in a timely and inexpensive manner. Kovacs, et al. [37], [38], [39], developed a plasma etching technique for the fabrication of via holes that overcomes the disadvantages of the laser-drilling technique initially used. Devices fabricated in this way appear to be compatible with the regeneration of physiologically viable axons [39].

In order to provide the low-impedance, high current density and long lifetime characteristics desired for the microelectrodes, iridium was chosen as the microelectrode material [40]. A technique for the fabrication of thin-film iridium microelectrode structures was subsequently developed [41].

Silicon nitride was chosen as a passivation layer to protect the microcircuits on the neural interface from degradation due to alkali ions present in body fluids. A low-temperature, plasma-enhanced chemical vapor deposition (PECVD) technique was used to provide the desired highly adherent and conformal coating.

Passive Chip Fabrication Process

Integration of the iridium deposition, passivation, and plasma etching processes was carried out in order to fabricate prototype passive neural interfaces (without active microelectronic circuits). Connections could be made only to some of the microelectrode sites within the arrays due to topological constraints (that can, as discussed above, be overcome by the addition of active circuits to scan the microelectrode sites). Upon completion of these process steps, the wafers were mechanically lapped to the desired thicknesses, on the order of 100 μm . Several microelectrode geometries and via hole sizes were designed and fabricated. One design of such passive neural interfaces is shown in Figure 12 below.

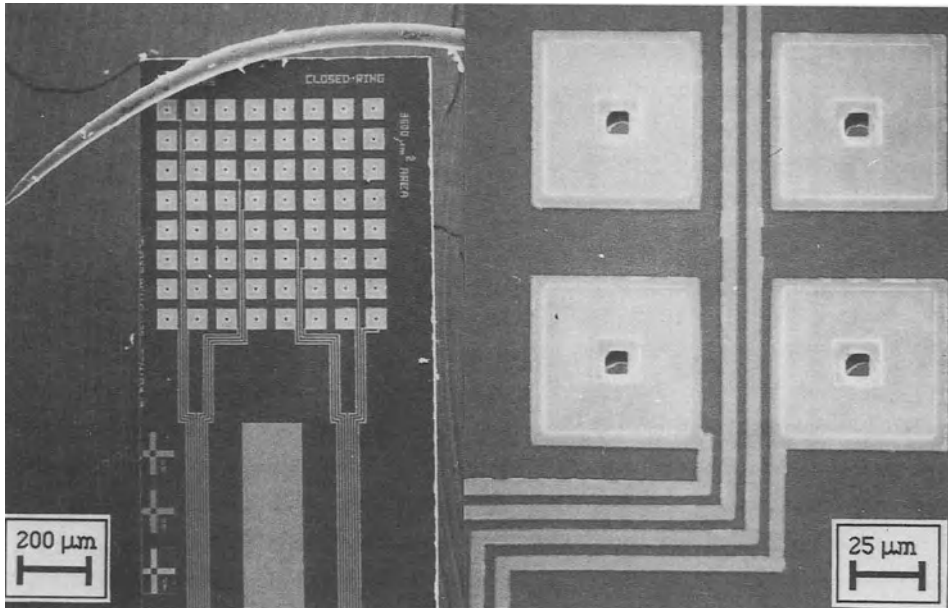


Figure 12: Two SEM views of a version of the passive neural interface showing iridium microelectrodes with central via holes (the left view also shows a microsurgery needle across the top left corner of the microelectrode array). The right view is a close-up of the microelectrodes showing the via holes in more detail. (magnifications = 50X and 400X, for left and right images, respectively)

Passive Chip *In Vivo* Studies

Passive neural interfaces were mounted in polycarbonate or resorbable GTMC (glycolide trimethylene carbonate) surgical couplers¹⁰ and interposed between the surgically severed ends of rat peroneal nerves. A completed assembly is shown in Figure 13 below. Preliminary electrophysiological studies using such interfaces implanted in Sprague-Dawley rats have been encouraging. Initial results indicate that both stimulating and recording with individual microelectrode sites is possible using devices implanted for over one year [42], [43]. Current work is focused on determining the selectivity of the microelectrodes and optimizing the density and size of the via holes.

¹⁰The surgical couplers were supplied by Davis and Geck, Inc., Pearl River, NY.

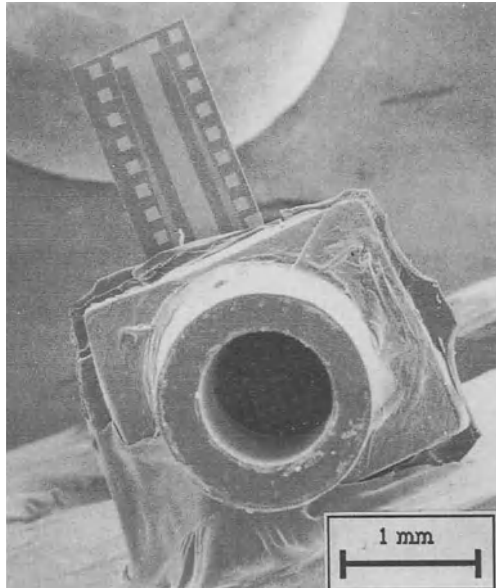


Figure 13: SEM view of a passive neural interface mounted in a polycarbonate surgical coupler. The severed ends of the nerve fascicle are inserted into the cylindrical openings on both sides of the surgical coupler (one of which faces forward in the image above) and they are sutured in place. (magnification = 22X)

Active Neural Interface Prototype

As explained above, active circuits within the interface itself are required in order to overcome the topological constraints on passive designs and to permit time-multiplexing of the neural signals to reduce the number of connections to and from the neural interface. A 32×32 microelectrode, X-Y addressable active neural interface was designed, followed by fabrication of the devices using $3.0 \mu\text{m}$ CMOS technology [44]. Figure 14 below is an SEM view of the prototype active neural interface. The interface permits either scanning of all 1,024 microelectrode sites sequentially or the selection of any individual site. In both modes, the device is bidirectional, permitting recording or stimulation.

This prototype was fabricated using a commercial vendor¹¹ who provided diced devices that could not be processed further (which would have required intact wafers) to make them suitable for implantation. However, the prototype has served as an useful test vehicle, permitting the development of novel non-contact testing techniques and the verification of the basic operational characteristics of the design (such as power dissipation, discussed below). Only five connections are required to the neural interface, permitting bidirectional signal flow to and

¹¹MOSIS, a low-cost integrated circuit fabrication service administered by USC/ISI, Inc., Marina del Rey, CA.

from an external controller and providing power to the device. Control of clocking and generation of real-time graphical displays of the recorded signals (or stimulation patterns) was accomplished using an external control system developed for this and future devices.

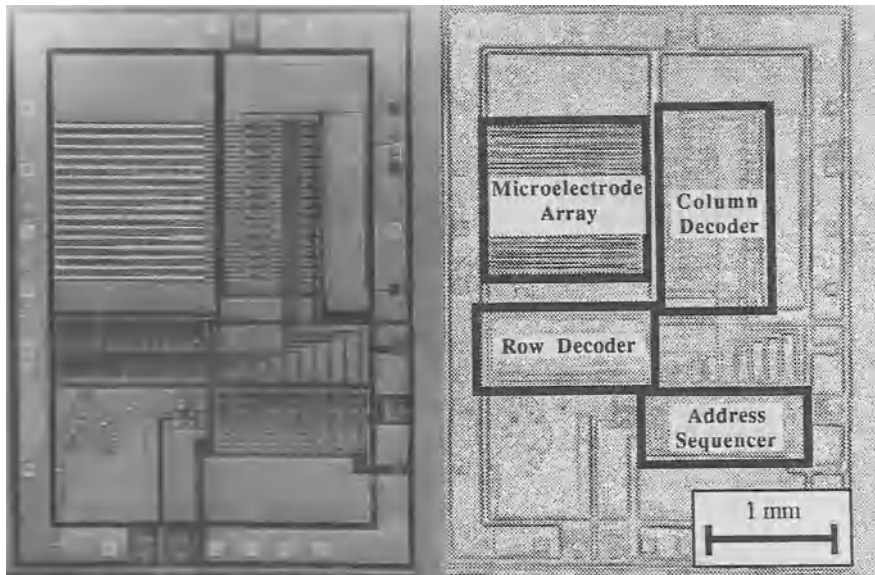


Figure 14: SEM view of the prototype active neural interface chip. The image at right is a high-pass filtered view showing the circuit blocks of the device. (magnification = 20X)

Power Dissipation Considerations for the Neural Interface

One of the major considerations in the design of the direct neural interface is power dissipation and the resulting chronic heating of the neural tissue. Thermal models are under development to allow for a design that will not overheat the nerve.

The prototype active neural interface device demonstrated a power dissipation of 1.8 mW at a sample frequency of 3 MHz [44]. This version of the device consists of a minimum of circuitry: 1,024 microelectrodes and transmission gates, row and column decoder circuits, an address sequencer, and various test structures. While the prototype was not fully optimized for low power dissipation, it did serve as a useful starting point. Work is currently being carried out to determine the *in vivo* power dissipation limits for active neural interfaces.

Use Of Neural Interfaces for Upper Limb Prostheses

Upper limb prostheses will be used as an example throughout this discussion of a proposed application for the neural interfaces. Current techniques use mechanical command signals from unaffected tissues (such as shoulder movements), or EMG signals from remaining musculature near the amputation site, to control the prosthesis. In *myoelectric* prostheses, EMG signals resulting from isometric contraction of muscles are recorded with surface electrodes and used to control prosthesis movements via signal-processing algorithms that decode “commands” to the mechanism. Problems commonly cited with myoelectric prostheses [45], [46] include lack of reliability of the EMG electrodes (e.g. susceptibility to faulty operation in the presence of perspiration), the need to concentrate constantly on the muscles used to maintain a grip, and the lack of any shear (slippage) force feedback¹². As well, inconsistent placement of the electrodes can make the requisite signal processing more difficult [46]. All of these problems reflect limitations of the interface between the patient and the prosthesis.

Because old-fashioned, purely mechanical “claw” devices provide proprioceptive feedback via their shoulder harnesses, they are often preferred by patients over more modern myoelectric prostheses [45]. Furthermore, Patterson and Katz [47] showed that even the use of a relatively simple feedback system (comprised of either a pressure cuff or electrocutaneous stimulation) in combination with visual feedback, improved the ability of test subjects to replicate grasping pressure over repeated trials. They also observed that with no visual feedback, the single-mode (pressure cuff) sensory feedback permitted better performance than did the mixed-mode (electrocutaneous stimulation) feedback. These results imply that a simple prosthesis that provides feedback to the user may be preferred over more technologically and esthetically advanced devices without feedback. The utility of such feedback schemes could be increased further if the feedback were obtained via interfaces to the *normal* control and sensory pathways of the limb.

Advanced mechanical and electronic technologies have allowed the evolution of reasonably dextrous robotic hands [48], as well as tactile and other types of sensors. From such devices, advanced motor/sensory prostheses could be constructed. All sensory modalities present in a natural limb (touch, vibration, joint position, force, and “pain” or damage) can now be sensed using solid-state transducers [49]. The dexterity of robotic limbs has gone far beyond that offered by “claw” prostheses. The major limiting factor preventing their clinical use is again the interface to the human operator.

Improvement of the interface between the patient and the prosthesis, resulting in the ability to use more advanced prostheses, thus appears to be the area where research could most benefit the patient. An ideal interface would allow use of the limb via the normal efferent and

¹²(which is normally used to correct the force of one’s grip to prevent the dropping of an object held in the hand)

afferent neural channels. The neural interface under development should allow the bidirectional transfer of information between a limb stump and the signal-processing electronics of the prosthesis. The signal-processing circuitry would interpret efferent firing patterns to control the motor functions of the prosthesis and map sensory information from transducers in the prosthesis into the afferent pathways.

Feasibility of Neural Interface Use for Prostheses

Regarding the feasibility of using direct neural interfaces in humans, a key question is whether or not the ability of the CNS to access the PNS signal paths to and from an amputation site is maintained over time. From discussions with amputation patients, who describe a gradual “telescoping” of phantom limbs “into” their stumps, it seems likely that some changes are occurring at the CNS level. However, Stein, et al. [50], [51], demonstrated in 1980 that stimulation of the ulnar nerve in the thirty-year-old stump (from a below-the-elbow amputation) of a human volunteer resulted in tingling sensations in the normal cutaneous innervation field of the ulnar nerve (small finger, ulnar half of ring finger, and ulnar third of hand). In 1982, De Luca, et al. [52], published encouraging preliminary results of the recording of compound action potentials from the median, ulnar, and radial nerves in a below-the-elbow amputation in conjunction with volitional commands by the patient. These results lend credence to the idea that CNS commands can be accessed in peripheral nerves in limb stumps, even long after the amputation is carried out.

Proposed Adaptive Prosthesis Control Strategy

The signals employed by the PNS to control various hand motions are complicated and case specific. Thus, in order to make use of the thousands of signals that should be available from the neural interfaces, it will be necessary to employ an adaptive system capable of utilizing the information content available in the signals and learning how the signals can be used to control a mechanical prosthesis. These requirements could be met through the use of artificial neural networks, as discussed in detail in a previous publication [53]. Thus, the majority of the burden involved in training will be placed on the prosthetic devices themselves, rather than the patient.

A block diagram of a complete prosthesis system is shown in Figure 15 below, followed by an artist’s conception of the physical construction of such a system, shown in Figure 16. For *efferent* signals, it would be necessary to perform some degree of feature extraction to reduce the overall data rate of the system, specifically to minimize the amount of data flowing from the neural interfaces to the neural network. This could be performed in several stages. Initial feature extraction would consist of demodulating the neural signals from each microelectrode site on the neural interface into numerical representations of their effective axonal firing rates. Within the stump, these demodulated signals would be multiplexed into a common

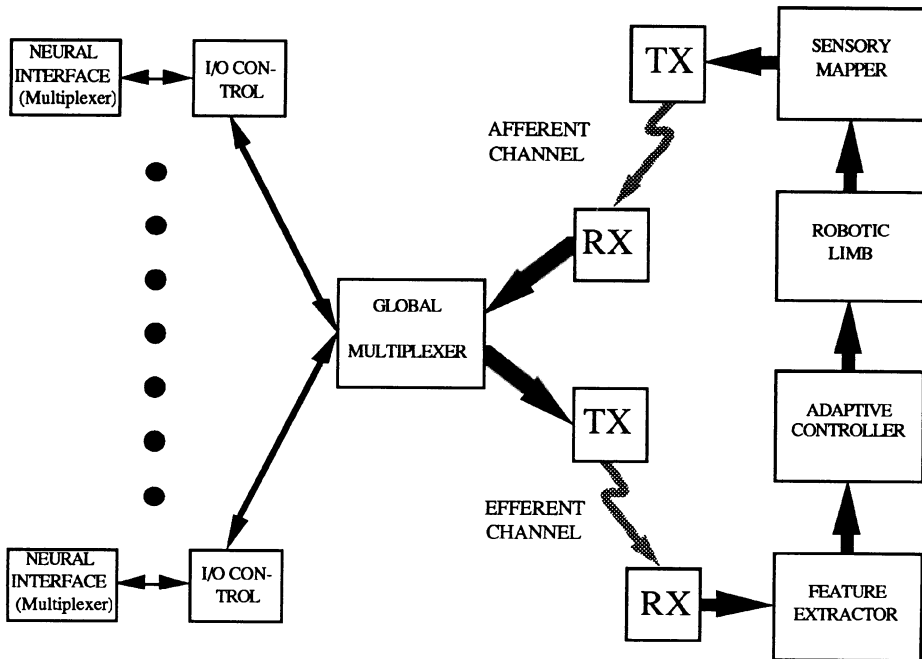


Figure 15: Block diagram of a directly interfaced limb prosthesis system. Signals to and from the neural interface for each fascicle are routed by input/output (I/O) controllers that either demodulate or stimulate as appropriate for each microelectrode site. The data for each fascicle flows through a global multiplexer prior to reaching the implanted transceiver. (Each transceiver is depicted as separate transmitter (TX) and receiver (RX) blocks and broken arrows denote transcutaneous transmission of information.) For the efferent channel, demodulated neural signals are processed by a feature extractor, followed by an adaptive neural network used to control the robotic limb. Information from transducers in the robotic limb is mapped onto the afferent channel by a neural network sensory mapper in order to provide sensory feedback.

signal and transmitted to the external prosthesis hardware via a telemetry system. Following this, an *adaptive feature extractor* would cluster signals into functionally similar groups and then form an average demodulated signal for each feature signal. This additional data reduction is necessary to reduce the complexity of further processing. Finally, downstream from the feature extractor, an adaptive neural network in the prosthesis would interpret the neural command signals and control the mechanical systems of the prosthesis.

For *afferent* information, signals would be processed in a similar manner, but in a reverse direction. Signals from transducers in the prosthesis would be mapped onto the appropriate sensory channels using a second adaptive network. The outputs of the adaptive network would consist of numerical representations of the desired stimulation rates at the microelectrodes. Signals from this network would then be multiplexed into a common signal and transmitted into the stump. This sensory information would be de-multiplexed and routed to the appropriate microelectrode sites, having been translated into the appropriate stimulation rates to represent it.

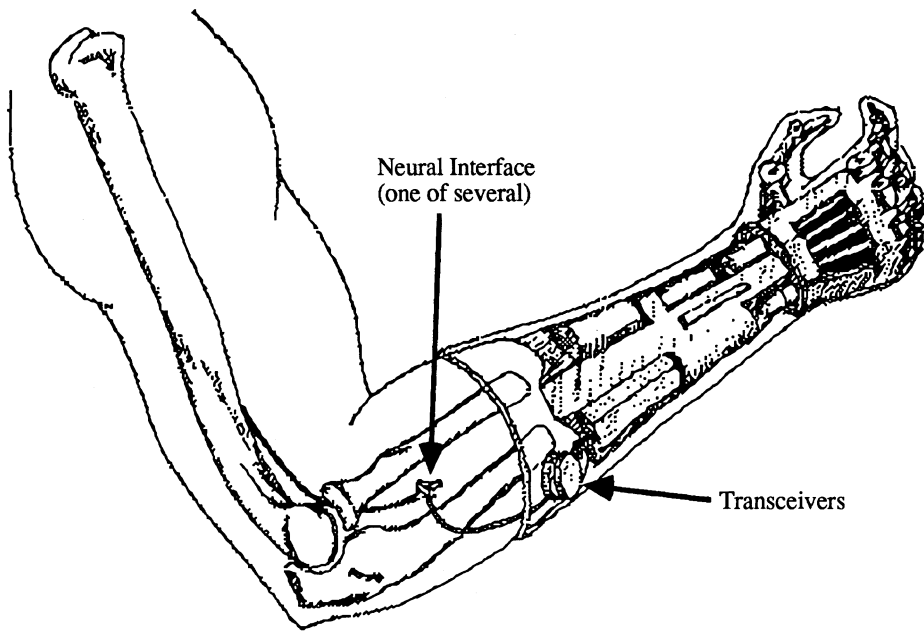


Figure 16: Artist's conception of a directly interfaced prosthetic hand showing a neural interface connected to an implantable transceiver in communication with its counterpart in the prosthesis. As currently envisioned, neural network control hardware would reside in the prosthesis. [Drawing by W. R. Knapp.]

Training Methodology for Adaptive Prosthesis

The following technique could be used to establish a basic set of neural command patterns with which to train the adaptive neural network to control the prosthesis and to sort axons into afferent or efferent groups simultaneously with this training. The patient would be asked to mimic, with his or her "phantom" hand, a predefined set of motions that could be presented using a computer-generated representation of a hand. Several records of the neural firing patterns corresponding to each motion would be stored for later use in "off-line" training of the feature extractor and neural network control system. It may also be useful to have the patient carry out a set of hand motions with a normal hand (if present), using a device such as the data glove to measure joint angles and positions in the state space of the hand. This would allow some of the motions to be defined by the patient to better suit his or her individual needs.

In order to separate afferent from efferent axons, the axonal signals that showed no consistent electrical activity during these tests would be classified as afferent. The individual microelectrode sites corresponding to them could then be defined as sites for stimulation if the prosthesis were equipped with sensory input transducers. The process for incorporating sensory capability in the prosthesis would require a separate procedure. While the characteristics

of the transducers on the prosthesis itself would be known, sorting out the different classes of afferent fibers (tactile sense, proprioception, pain, and temperature sense) is likely to be a much more arduous task. The major difficulty would be that when presented with various stimuli, the patient would somehow have to report his or her perceptions to the sorting algorithm. A proposed method for training sensory information into the prosthesis that also utilizes an adaptive neural network has been previously presented [53].

Anatomical Considerations

A prosthesis built with such technology would have to be extremely flexible in terms of adaptability to the needs and anatomical constraints of the patient. Since the exact site of the injury and neural, muscular, and skeletal damage would be variable, the prosthesis would have to be adaptable both mechanically and electronically. Mechanical adaptability would be handled via a range of physical options (necessitated by the range of amputation site sizes and degrees of injury) to *physically* interface the prosthesis hardware to the patient and the electronic adaptability, as discussed above, to *functionally* interface the entire system to the patient.

The complex microsurgery required for the implantation of the neural interfaces will no doubt be limited in some instances by cost factors, emphasizing the importance of identification of only the appropriate fascicles. Such identification could potentially be done under carefully applied local anesthesia by asking the patient to mimic pre-defined hand movements (as described above for training of the neural network).

Intraneural cross-section maps [54], [55], [56] will undoubtedly be useful for *estimating* the location of the desired fascicles within each nerve and the number of neural interfaces required, depending upon the level of the injury and the capabilities of the prosthesis. However, since individual anatomical variations may impede the identification of only those fascicles relevant to the available prosthesis, it is likely that many of the accessible fascicles would be outfitted with neural interfaces in anticipation of more advanced prostheses. Only the interfaces carrying information relevant to the level of complexity of the prosthesis would be activated once they were identified (in order to minimize the power and signal processing required).

As an example of a prosthesis for a below-the-elbow amputation (as illustrated in Figure 16 above), it can be assumed that of the elbow flexors at least the *biceps brachii* and *brachialis*¹³ and the elbow extensors *triceps brachii* and *anconeus* would be spared. It is likely that any useful pronation/supination function would be absent (primarily due to the fact that at present mechanical coupling to the remains of the radius and ulna would be difficult). The three major nerves in the arm, the radial, median, and ulnar, could in most cases be readily located surgically. Given an amputation site without medical complications, microsurgical implantation of the neural interfaces could be undertaken in these nerves. It is clear that for a “complete” hand prosthesis, neural interfaces would be required for all fascicles of these nerves.

¹³(the *brachioradialis* would likely be lost since it is mainly located in the forearm)

Considering simpler hand prostheses, it is felt that the most important function would be thumb opposition to a digit or digits. This is, in fact, the only function of the basic “claw” prostheses. For this “simplest case” prosthesis, a way of obtaining this function would be to locate and utilize the motor information that would have been routed to the *opponens pollicis* (or either of *flexor pollicis longus* or *brevis*) muscle for initiating opposition and either of the *extensor pollicis* muscles (*brevis* or *longus*) for release. Sensory information for the dorsal surface of the thumb is carried by the radial nerve and for the fingertips in the median (index and adjacent finger) and the ulnar nerve (two medial fingers). Thus, it is clear that even for the most basic prosthesis, interfaces must be implanted in some fascicles of all of the three major nerves supplying the muscles and skin of the forearm.¹⁴

Comments on Hardware Implementation

Robotic hands have been designed with dexterity comparable to that of the human hand (the Utah/MIT hand for example [57]). However, completely self-contained units with comparable dimensions to those of human limbs do not presently exist. Prostheses with reduced functionality but similar dimensions to natural limbs are available, as evidenced by myoelectric prostheses. While it is beyond the current scope of this research project to develop more functionally advanced and compact devices, the general trend in robotics and prosthetics has been in this direction.

Suitable telemetry systems for bidirectional transmission of data in this application appear to be achievable with present technologies. For example, the simultaneous transmission of data and power via high-frequency electromagnetic coils has been demonstrated in prosthetic applications [58].

With respect to the signal processing discussed above, integrated circuit implementations of the required neural networks are becoming available [59]. Thus, it is likely that off-the-shelf components could be used for much of the electronics with the exception of the neural interfaces themselves. While some fabrication steps of the neural interfaces will likely remain laboratory processes, the use of inexpensive commercial processes where possible has been emphasized. Further, in order to achieve the goal of a realistic and cost-effective prosthesis, an active effort is being made to avoid the use of expensive and esoteric materials and fabrication processes in all aspects of the project.

Additional Applications for Neural Interfaces

Additional uses for the neural interfaces and processing circuitry would be abundant. Once a prosthesis interface was established, the processing circuitry could be trained for alternative devices that could be connected to the patient. Thus the neural interface and associated circuits would constitute an extremely versatile man/machine interface.

¹⁴For a “complete” forearm prosthesis, the cutaneous input of the musculocutaneous nerve over the radial forearm must also be considered.

Control of mechanical devices could be accomplished without the mechanical lag of the hand. For example, the control surfaces of an aircraft could be directly mapped to hand motion commands. Sensory ranges could be compressed or expanded to suit many applications. Microscopic manipulations, such as those of microsurgery, could be mapped into perceived macroscopic motions. New sense modalities could also be introduced. For example, radiation could be sensed using the appropriate transducers and mapped into temperature sense in the operator. It should be noted that since information is bidirectionally transmitted into and out of the neural interfaces in a form suitable for radio transmission, remote operation would also be inherently possible in these and other applications.

In addition, these devices will allow for a great deal of basic science research that should answer fundamental questions regarding the nature of information conveyed by the PNS.

Conclusions

This article has presented an overview of man/machine interfaces, the history of the development of regeneration-type neural interfaces, the neural interface technology under development by this group, and a proposed implementation of a directly interfaced prosthesis. A long-term, multi-disciplinary project is underway in an effort to realize such prostheses. It is expected that it will be on the order of a decade before clinically useful devices can be produced. In the meantime, it is hoped that some of the technologies developed in the course of the overall project will find uses in rehabilitation and basic science research.

Addendum

There was a lengthy delay (nearly three years) between the submission of this article and its publication. We have made no effort to update the material to include our latest research results, some of which are of considerable importance in terms of demonstrating the basic principles discussed herein.

G. T. A. Kovacs, September 14th, 1992

The interested reader is referred to the following article for further information:

Kovacs, G. T. A., Stormont, C. W., and Rosen, J. M., "Regeneration Microelectrode Array for Peripheral Nerve Recording and Stimulation," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 9, pp. 893-902, Sept. 1992.

Acknowledgments

This research has been funded under a Department of Veterans Affairs (DVA) Rehabilitation Research and Development (RR&D) Merit Review Grant (B003 Hentz/Rosen "Towards Better Methods of Nerve Repair and Evaluation"). The authors wish to extend their sincere thanks to Laurel Joyce of the Stanford University Office of Communications for her editorial comments and careful review of this work, to Doug Franz of Hewlett-Packard, Inc., for his constant enthusiasm and intellectual input to the project, to Chris Storment of the DVA for his ongoing efforts to realize the fabrication technologies required and for providing many of the scanning electron micrographs presented herein, to Khoi Nguyen of the DVA for his assistance with the microsurgical procedures, to Nancy Kelly and Joanne Kipp (both of the DVA) for their help in producing the SEM's and TEM's of biological tissues, to Kate Clark and Woody Knapp (both of Stanford University) for their beautiful artwork, to Martin Brooks of the Senter for Industri-forskning, Oslo, Norway, for his comments on this paper, and to our many friends and collaborators who have "lent us a hand" with the research.

References

- [1] S. Brand, "The Media Lab," New York: Viking Penguin, Inc., 1987.
- [2] R. W. Mann, "Cybernetic Limb Prosthesis: The ALZA Distinguished Lecture," *Annals of Biomedical Engineering*, vol. 9, pp. 1-43, 1981.
- [3] R. B. Knapp and H. S. Lusted, "Bioelectrical Controller for Computer Music Applications," *Computer Music Journal*, vol. 14, no. 1, pp. 42-47, Spring 1990.
- [4] E. Sutter, "Communication Aid Utilizing Brain Responses," *Proceedings of the 3rd Annual Conference on Computer Technology/Special Education/Rehabilitation*, California State University, Northridge, CA, Oct. 1987, pp. 487-498.
- [5] L. L. Baker, C. Yeh, D. Wilson, and R. L. Waters, "Electrical Stimulation of Wrist and Fingers for Hemiplegic Patients," *Physical Therapy*, vol. 59, pp. 1495-1499, 1979.
- [6] D. Popovic, R. Tomovic, and L. Schwirtlich, "Hybrid Assistive System—The Motor Neuroprosthesis," *IEEE Transactions on Biomedical Engineering*, vol. 36, no. 7, pp. 729-737, July 1989.
- [7] J. Petrofsky, "Muscle Fiber Recruitment and Blood Pressure Response to Isometric Exercise," *Journal of Applied Physiology*, vol. 50, no. 1, pp. 32-37, Feb. 1981.
- [8] E. Tyler, C. Caldwell, and J. N. Ghia, "Transcutaneous Electrical Nerve Stimulation: An Alternative Approach to the Management of Postoperative Pain," *Anesthesia and Analgesia*, vol. 61, pp. 449-455, 1982.
- [9] H. Granek and M. Granek, "Transcutaneous Transducer Garments: Current Mammalian Usage and Future Applications," *The Journal of Neurological and Orthopaedic Medicine and Surgery*, vol. 6, no. 3, pp. 271-278, Oct. 1985.
- [10] P. H. Peckham, J. T. Mortimer, and E. B. Marsolais, "Controlled Prehension and Release in the C5 Quadriplegic Elicited by Functional Electrical Stimulation of the Paralyzed Forearm Musculature," *Annals of Biomedical Engineering*, vol. 8, pp. 369-388, 1980.

- [11] P. H. Peckham, C. W. Poon, W. H. Ko, E. B. Marsolais, and J. J. Rosen, "Multichannel Implantable Stimulator for Control of Paralyzed Muscles," *IEEE Transactions on Biomedical Engineering*, vol. BME-28, pp. 530-536, 1981.
- [12] E. B. Marsolais, "Electrical Activation of Muscles for Open-Loop Walking," *Proceedings of the 5th IEEE Engineering in Medicine and Biology Conference*, Columbus, OH, 1983.
- [13] R. L. White, "Review of the Current Status of Cochlear Prostheses," *IEEE Transactions on Biomedical Engineering*, vol. BME-29, no. 4, pp. 233-238, April 1982.
- [14] P. A. Leake, D. K. Kessler, and M. M. Merzenich, "Application and Safety of Cochlear Prostheses," Chapter 10 in *Neural Prostheses*, W. F. Agnew and D. B. McCreery, Eds., Englewood Cliffs, NJ:Prentice Hall, 1990, pp. 254-296.
- [15] G. S. Brindley, and W. Lewin, "The Sensations Produced by Electrical Stimulation of the Visual Cortex," *Journal of Physiology*, vol. 196, pp. 479-493, 1968.
- [16] D.J. Anderson, K. Najafi, S. J. Tanghe, D. A. Evans, K. L. Levy, J. F., Hetke, X. Xue, J. J. Zappia, and K. D. Wise, "Batch-Fabricated Thin-Film Electrodes for Stimulation of the Central Auditory System," *IEEE Transactions on Biomedical Engineering*, vol. 36, no. 7, pp. 693-704, July 1989.
- [17] K. Najafi, J. Ji, and K. D. Wise, "Scaling Limitations of Silicon Multichannel Recording Probes," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 1, pp. 1-11, Jan. 1990.
- [18] G. G. Naples, J. T. Mortimer, and T. G. H. Yuen, "Overview of Peripheral Nerve Electrode Design and Implantation," Chapter 5 in *Neural Prostheses*, W. F. Agnew and D. B. McCreery, Eds., Englewood Cliffs, NJ: Prentice Hall, 1990, pp. 108-145.
- [19] J. A. Hoffer, M. Haugland, and T. Li, "Obtaining Skin Contact Force Information From Implanted Nerve Cuff Recording Electrodes," *Proceedings of the 11th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Seattle, WA, Nov. 1989, pp. 928-929.
- [20] K. D. Wise, J. B. Angell, and A. Starr, "An Integrated-Circuit Approach to Extracellular Microelectrodes," *IEEE Transactions on Biomedical Engineering*, vol. BME-17, no. 3, pp. 238-247, July 1970.
- [21] K. D. Wise and J. B. Angell, "A Low-Capacitance Multielectrode Probe for Use In Extracellular Neurophysiology," *IEEE Transactions on Biomedical Engineering*, vol. BME-22, p. 212-219, 1975.
- [22] R. S. Pickard, "A Review of Printed Circuit Microelectrodes and Their Production," *Journal of Neuroscience Methods*, vol. 1, pp. 301-318, 1979.
- [23] S. E. MacKinnon and A. L. Dellon, "Surgery of the Peripheral Nerve," New York: Thieme Medical Publishers, Inc., 1988.
- [24] A. F. Marks, "Bullfrog Nerve Regeneration into Porous Implants," *Anatomical Record*, vol. 163, p. 226, 1969.
- [25] A. F. Foerster (formerly Marks), personal communication.
- [26] R. Llinás, C. Nicholson, and K. Johnson, "Implantable Monolithic Wafer Recording Electrodes for Neurophysiology," Chapter 7 in *Brain Unit Activity During Behaviour*, M. Phillips, Ed., Springfield, IL:Thomas, 1973, pp. 105-111.
- [27] A. Mannard, R. B. Stein, and D. Charles, "Regeneration Electrode Units: Implants for Recording from Peripheral Nerve Fibers in Freely Moving Animals," *Science*, vol. 183, pp. 547-549, Feb. 8, 1974.
- [28] G. E. Loeb, W. B. Marks, and P. G. Beatty, "Analysis and Microelectronic Design of Tubular Electrode Arrays Intended for Chronic, Multiple Single-Unit Recording from Captured Nerve Fibers," *Medical & Biological Engineering & Computing*, vol. 15, pp. 195-201, March, 1977.
- [29] K. E. Peterson, "Silicon as a Mechanical Material," *Proceedings of the IEEE*, vol. 70, no. 5, pp. 420-457, May 1982.
- [30] T. Matsuo, A. Yamaguchi, and M. Esashi, "Fabrication of Multi-Hole-Active Electrode for Nerve Bundle," *Journal of Japan Soc. ME & BE*, July 1978.

- [31] N. Hoshimiya, Department of Electrical Communications, Tohoku University, personal communication.
- [32] D. J. Edell, "Development of a Chronic Neuroelectronic Interface," Doctoral Dissertation, U. C. Davis, 1980.
- [33] D. J. Edell, L. D. Clark, and V. M. McNeil, "Optimization of Electrode Structure for Chronic Transduction of Electrical Neural Signals," *Proceedings of the 9th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Texas, Nov. 1986.
- [34] J. M. Rosen, M. Grosser, and V. R. Hentz, "Nerve Repair at the Axon Level: Long Term Regeneration of Axons in Rats and Primates Through Laser-Drilled Holes in Silicon Chips," submitted for publication.
- [35] T. R. Anthony, "Diodes Formed by Laser Drilling and Diffusion," *Journal of Applied Physics*, vol. 53, no. 12, pp. 1170-1175, Dec. 1982.
- [36] T. R. Anthony, "The Random Walk of a Drilling Laser Beam," *Journal of Applied Physics*, vol. 51, no. 2, pp. 9154-9164, Feb. 1980.
- [37] G. T. A. Kovacs, M. Stephanides, W. R. Knapp, J. P. McVittie, V. R. Hentz, and J. M. Rosen, "Development of Chronic Implant Neural Prosthesis Microelectrode Arrays," *Proceedings of the IEEE Montech Conference on Biomedical Technologies*, Montréal, PQ, Nov. 1987, pp. 152-155.
- [38] G. T. A. Kovacs, C. W. Storment, J. P. McVittie, W. R. Knapp, V. R. Hentz, and J. M. Rosen, "Advances Toward Development of Microelectronic Axonal Interface Neuroprostheses," *Proceedings of the International Conference of the Association for the Advancement of Rehabilitation Technology*, Montréal, PQ, June 1988, pp. 362-363.
- [39] G. T. A. Kovacs, C. W. Storment, K. D. Nguyen, M. Stephanides, J. P. McVittie, V. R. Hentz, and J. M. Rosen, "Micromachined Silicon Neural Implants: Development of Chronic Interfaces to Peripheral Nerves," submitted to *IEEE Transactions on Biomedical Engineering*.
- [40] L. S. Robblee, J. L. Lefko, and S. B. Brummer, "Activated Ir: An Electrode for Reversible Charge Injection in Saline Solution," *Journal of the Electrochemical Society*, vol. 130, pp. 731-733, Mar. 1983.
- [41] G. T. A. Kovacs, C. W. Storment, V. R. Hentz, and J. M. Rosen, "Fabrication Techniques for Directly Implantable Microelectronic Neural Interfaces," *Proceedings of the 12th Annual Conference of the Association for the Advancement of Rehabilitation Technology*, New Orleans, LA, June 1987, pp. 675-677.
- [42] G. T. A. Kovacs, Doctoral Dissertation in the Department of Electrical Engineering, Stanford University, in preparation.
- [43] G. T. A. Kovacs, C. W. Storment, T. K. Whitehurst, K. D. Nguyen, M. Stephanides, J. P. McVittie, V. R. Hentz, and J. M. Rosen, "Development and *In Vivo* Testing of Thin Film Iridium Microelectrode Arrays for Chronic Neural Interfaces," in preparation.
- [44] G. T. A. Kovacs, M. Stephanides, W. R. Knapp, J. P. McVittie, and J. M. Rosen, "Design of Two-Dimensional Neural Prosthesis Microelectrode Arrays," *Proceedings of the 9th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, MA, Nov. 1987, pp. 1651-1652.
- [45] D. Franz, personal communication.
- [46] C. J. De Luca, "Control of Upper-Limb Prostheses: A Case For Neuroelectric Control," *Journal of Medical Engineering Technology*, vol. 2, no. 2, pp. 57-61, Mar. 1978.
- [47] P. E. Patterson and J. A. Katz, "A Sensory Feedback System for Grasping Pressure in a Myoelectric Hand," *Proceedings of the IEEE Engineering in Medicine and Biology Society 10th Annual International Conference*, New Orleans, Nov. 1988, pp. 1564-1565.
- [48] S. C. Jacobsen, D. F. Knutti, R. T. Johnson, and H. H. Sears, "Development of the Utah Artificial Arm," *IEEE Transactions on Biomedical Engineering*, vol. BME-29, no. 4, pp. 249-269, April 1982.
- [49] W. H. Ko, "Solid-State Physical Transducers for Biomedical Research," *IEEE Transactions on Biomedical Engineering*, vol. BME-33, no. 2, pp. 153-162, Feb. 1986.

- [50] R. B. Stein, D. Charles, J. A. Hoffer, J. Arsenault, L. A. Davis, S. Moorman, and B. Moss, "New Approaches to Controlling Powered Arm Prostheses, Particularly by High-Level Amputees," *Bulletin of Prosthetics Research*, vol. BPR 10-33, pp. 51-62, 1980.
- [51] R. B. Stein, T. Gordon, J. A. Hoffer, L. A. Davis, and D. Charles, "Long-Term Recordings from Cat Peripheral Nerves During Degeneration and Regeneration: Implications for Human Nerve Repair and Prosthetics," in *Nerve Repair: Its Clinical and Experimental Basis*, D.L. Jewett and H.R. McCarroll, Eds., St. Louis, MO: C.V. Mosby, 1980, pp. 166-176.
- [52] C. De Luca, L. D. Gilmore, L. J. Bloom, S. J. Thomson, A. L. Cudworth, and M. J. Glimcher, "Long-Term Neuroelectric Signal Recording from Severed Nerves," *IEEE Transactions on Biomedical Engineering*, vol. BME-29, no. 6, pp. 393-402, June 1982.
- [53] E. A. Wan, G. T. A. Kovacs, J. M. Rosen, and B. Widrow, "Development of a Neural Network Interface for Direct Central Nervous Control of a Prosthetic Limb," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., (presented as a plenary paper by Dr. Bernard Widrow).
- [54] S. Sunderland, *Nerves and Nerve Injuries*, Edinburgh: Churchill Livingstone, 1972.
- [55] S. Sunderland, "Intraneural Topography of the Radial, Median and Ulnar Nerves," *Brain*, vol. 68, no. 243, pp. 17-299, 1945.
- [56] M. E. Jabaley, W. H. Wallace, and F. R. Heckler, "Internal Topography of Major Nerves of the Forearm and Hand: A Current View," *Journal of Hand Surgery*, vol. 5, no. 1, pp. 1-18, Jan. 1989.
- [57] S. Jacobsen, E. Iverson, et al., "Design of the Utah/MIT Dextrous Hand," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, 1986, pp. 1520-1532.
- [58] D. C. Galbraith, "An Implantable Multichannel Neural Stimulator," Doctoral Dissertation, Stanford University Department of Electrical Engineering, Technical Report No. G908-2, Dec. 1984.
- [59] M. Holler, S. Tam, H. Castro, and R. Benson (all of Intel Corporation), "An Electrically Trainable Artificial Neural Network (ETANN) With 10240 'Floating Gate' Synapses," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., June 1989, pp. 191-196.

Integrated Bioelectronic Transducers

M. Grattarola, A. Cambiaso, S. Cenderelli, S. Martinoia, M.T. Parodi, and M. Tedesco

Bioelectronics Lab, Biophysical and Electronic Engineering Department, University of Genoa, Via Opera Pia 11a - 16145 Genoa, Italy

1. Introduction

The characterization of single-unit activity from excitable cells is a key point in the study of information processing mechanisms involving correlated activity within cell populations. This knowledge represents a fundamental step for both basic research areas (neural computation) and clinical applications (functional prostheses). Recent advances in microelectronic techniques have made it possible to monitor on line electrochemical signalling among single units in a network of neurons grown, in a controlled way, under cell culture conditions. Several distinct parameters are relevant to the achievement of this goal, namely : a) Cytometric characterization of in vitro cell coupling to integrated devices; b) cell placement on the substrate according to defined patterns; and c) recording of electrical and chemical signals by means of integrated microdevices. These three parameters will be considered separately, with reference to the literature and to results obtained in our Lab.

2. Cytometric characterization of cell adhesion to integrated devices

Cell culture techniques provide a powerful tool for answering fundamental neurobiological questions. The biochemical environment is easily manipulated and relatively simple cell networks can be formed in which individual cells are visible and accessible. Under these experimental conditions, cytometric techniques can be used to characterize cell adhesion and morphology. They can also be used to monitor cell activity. Among cytometric techniques, digital epifluorescence microscopy is one of the most appropriate for these purposes. In fact, optical recording of electrochemical signalling from multiple single units can be obtained with voltage- and ion-sensitive dyes.

In the case of voltage-sensitive dyes, a small transient variation in fluorescence intensity can be correlated with the propagation of action potentials (1). The detection of this phenomenon is beyond the capability of present TV cameras and "ad hoc" detectors (e.g., matrices of photodiodes) must be used.

Ion-sensitive dyes represent powerful tools for imaging the space propagation of chemical signals within single units of a cell network. For example, the dye Fura-2, which can

be used for detecting intracellular free Ca^{++} (2), has been widely used for digital imaging of neurons and muscle cells (3,4).

Figure 1 shows a digital image obtained with chick embryo cardiac cells cultured on Al_2O_3 . This insulator was chosen because, (together with Si_3N_4), it is one of the most frequently used in ISFET devices (Cf. following sections). Cells were stained with the dye Fura-2 and images were obtained with an epifluorescence microscope equipped with a Xenon lamp. The microscope was interfaced with a double intensification ISIT TV camera. The images generated by the TV camera were digitized and saved on the floppy disk of a personal computer. The image in figure is a "ratio" image, obtained by dividing point by point an image excited at 350 nm by another image of the same object obtained at 380 nm, according to the "ratio imaging" approach (2).

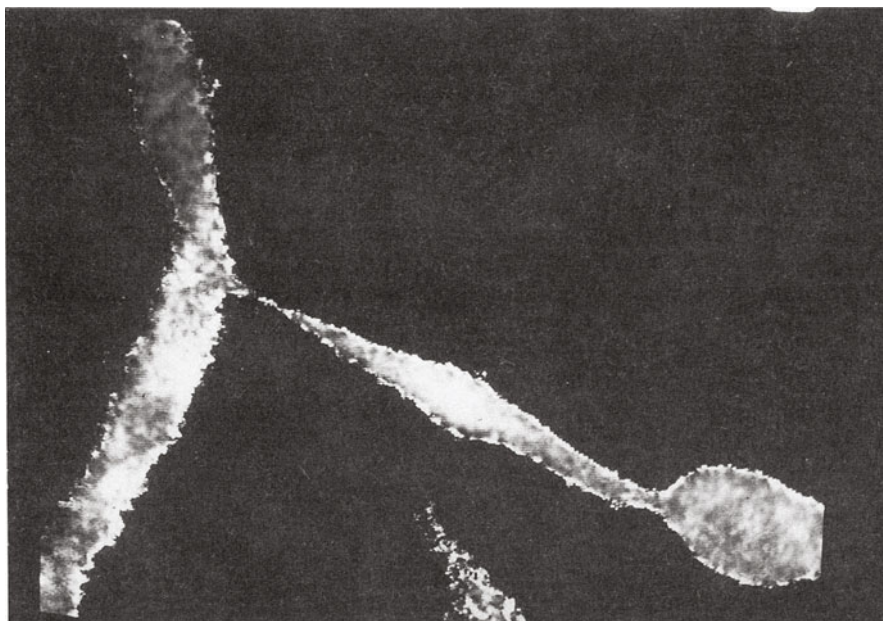


Fig. 1 Epifluorescence digital image of chick embryo cells. For more details see the text.

Another recent cytometric technique is represented by acoustic microscopy. High resolution scanning acoustic microscopy is very useful in imaging both microelectronic devices and biological materials in a non-destructive way (5,6). Variations in the local elastic properties of the sample under inspection affect the intensity of the reflected acoustic waves,

and generate the contrast in the reflected signal. The technique is especially useful in viewing the adhesion of unstained cells to opaque materials.

Figure 2 shows a digitized image of an unstained neuroblastoma cell obtained with the scanning acoustic microscope ELSAM (Leitz). The image was obtained at 1.6 GHz, using the culture medium as the coupling liquid between the lens and the cell. The focal plane is near the surface of the substratum (Si_3N_4). Small dark spots visible near the cell periphery could be interpreted as sites of strong adhesion (7).

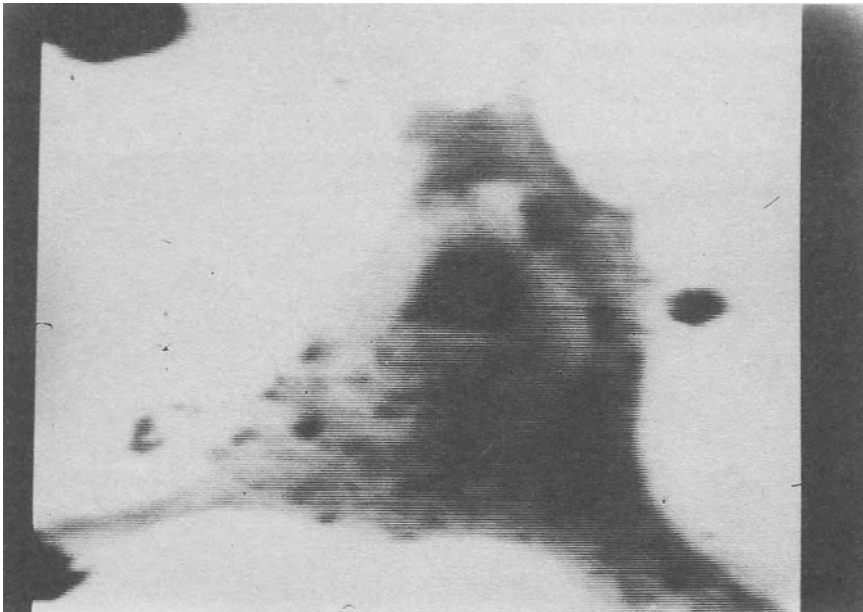


Fig. 2. Acoustic microscopy digital image of a neuroblastoma cell adhering to a laminin-coated substratum. Lateral resolution about $0.7\text{ }\mu\text{m}$.

3. Cell patterning

In order to couple single in vitro units of a cell network to single recording elements of a device array, cells and cell arborizations need to be guided to specific sites on the surface of the substratum. Recent advances in microelectronic techniques have made it possible to achieve this task. In fact, by using photolithographic techniques, the topography, adhesiveness and chemistry of synthetic surfaces can be modified at the μm scale. In this way, artificial

substrata with specific properties can be fabricated at the micrometer and sub-micrometer scales.

A very simple example of this approach is given in figure 3. Wafers of silicon covered by Al_2O_3 were coated with laminin, a glycoprotein known to favour the adhesion and differentiation of neuronal cells (8). The coated wafers were covered with circular golden masks with gratings of a few hundreds μm , and then exposed to UV light. In this way laminin was inactivated everywhere, except for the areas covered by the grid (9). These areas became preferential regions for cell adhesion. Figure 3 is a low magnification image showing different areas of the Al_2O_3 surface. Cells are identifiable as small dots outlining the circular border and the internal segments of a grid. This data is considered preliminary and a detailed review of this topic can be found in (10).

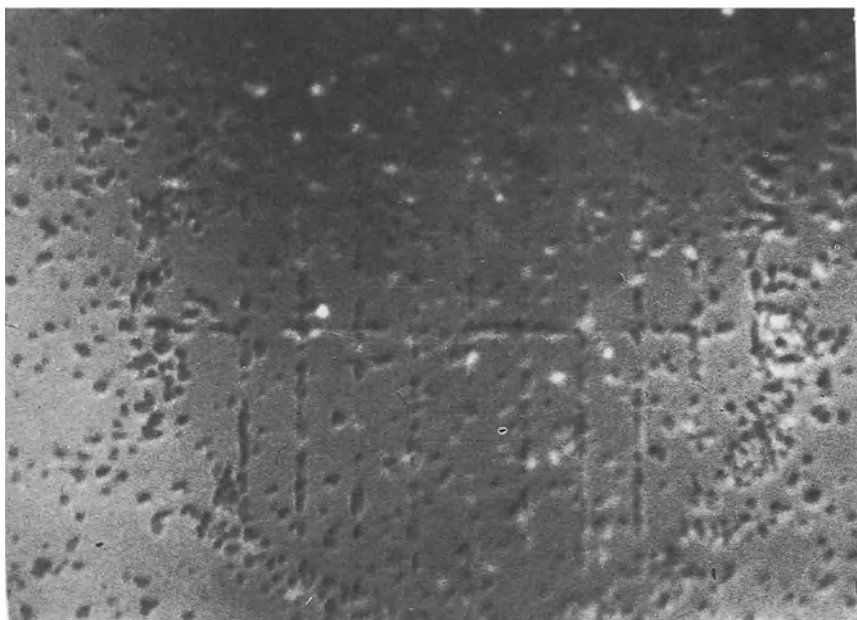


Fig. 3 Mouse neuroblastoma cells (low magnification) growing on patterned substrata.

4. Extracellular detection of cell activity.

The need for electrochemical characterization of a cell assembly implies a long-term (i.e., several hours) recording from many (up to several tens) individual cells. These requirements

dictate the use of extracellular vs. intracellular recording. Moreover, the intimate relationship between electrical phenomena and chemical messengers (both ions and neurotransmitters) stresses the importance of detecting a variety of molecular species at the cellular level. These considerations led to the design of an array of integrated microbiosensors functionally coupled to a network of excitable living cells. Until now, extracellular recording has been synonymous with electrical activity recordings. The use of metallic (Pt, Au) electrodes is a well-established technique for *in vivo* extracellular recording (11). Recently, the development of electrode microfabrication techniques has made this approach well-suited to recording from single excitable cells in a cell culture (12,13). The size of the microelectrodes can be as small as $10 \times 10 \mu\text{m}$, with ultimate size limited only by an increase in impedance. Using microphotolithographic techniques, arrays of tens of microelectrodes regularly spaced over an area of a few millimeters, are easily obtained. Moreover, as already discussed, geometrical and/or chemical modifications of the substrate surface can guide cells and cell arborizations toward selected sites (10). Metallic electrodes are (mostly) capacitively coupled to cells. They passively transduce the action potential of a cell into a signal which is then amplified. "Active" electrode arrays, which incorporate their own amplifiers, have been built and tested (14). In this case, each microelectrode is part of the gate electrode of an underlying MOS (metal - oxide - semiconductor) device, which is used in the source follower mode as a buffer amplifier. A miniature multiple thin-film recording sensor was recently used to simultaneously measure the electrical activity, oxygen content, and temperature of brain tissue (15).

Another line of research dealing with silicon based integrated biosensors, has resulted in the development of ion and chemical sensors based on the field effect transistor (FET) and, more recently, on passive silicon devices (LAPS, light addressable potentiometric sensor, (16)). An early publication by Bergveld (17) pointed out the possibility of using Ion Sensitive Field Effect Transistors (ISFETs) to record from electrically active cells. We shall conclude this section by discussing how an ISFET can detect electrochemical signals generated by excitable cells.

A firing cell, placed directly against the unmetallized gate insulator of an ISFET, can capacitively modify the behaviour of the device by transiently inducing electrical currents into and out of the cell membrane. This transient current is generated by a capacitive term and by local ion fluxes, mainly Na^+ and K^+ . Depending on whether or not the insulating layer has specific binding sites for one of these ions (18), modulation of the device will be regulated by the concentration of specific ions or simply by the total charge variation. When considering insulating layers with binding sites for H^+ only, (e.g. Si_3N_4 or Al_2O_3), the "non-specific" (i.e., related only to the net ionic charge) response of the system to the electrical component of the perturbation will be analyzed. It should also be noted that, if conditions for a stable seal between the cell and the insulator could be satisfied, then a situation of "loose patching" (19)

could be approximated. A subsequent local rupture of the cell membrane could then produce a situation approximating intracellular recording.

Starting from the core conductor model (20), it can be shown that the extracellular potential corresponding to an action potential travelling a long cylinder is given by :

$$V(x,y,z) = \frac{a^2 \sigma_i}{4 \sigma_e} \int_0^\infty \frac{\partial^2 \varphi_{in}}{\partial s^2} \frac{1}{\sqrt{(x-s)^2 + y^2 + z^2}} ds \quad [1]$$

where φ_{in} is the action potential, a is the fiber cross section and σ_i and σ_e are the intracellular and extracellular conductivities, respectively. φ_{in} can be approximated by using an analytical function mimicking the experimental data, as follows :

where A , λ and B are parameters chosen to simulate the action potential of a muscle fiber. A slightly modified expression can be used for nerve fibers (20).

By introducing the expression [2] into [1], by transforming x into $x + x_0 + vt$, (where v is the propagation speed of the action potential (0.5 - 100 m/s)), and by integrating over x and y (dimensions of the gate insulator), an expression approximating extracellular recording for fibers in contact with the device (i.e., $z = 10$ nm) can be found. This expression was utilized

$$\varphi_{in} = \begin{cases} A x^3 \exp(-\lambda x) - B & x \geq 0 \\ -B & x < 0 \end{cases} \quad [2]$$

as an input to the ISFET, and modelled according to standard approximations (21). The MOSFET was simulated to work in the linear region of its $I - V$ curve. Standard geometrical parameters were utilized for the device . Simulation programs were written in FORTRAN 77 for an HP 9000/360 computer. The results are depicted in figure 4, where the current I_D is given as a function of time for different values of the distance between biological tissue and a recording device (z coordinate) . At 10 nm from the fiber surface and for extracellular potentials on the order of hundreds of μV , a current of several tens of μA can flow in a

standard device with a gate insulator measuring $5\text{ }\mu\text{m} \times 50\text{ }\mu\text{m}$.

The above approximation is appropriate for a long fiber. More complex patterns, such as networks of neurons, need more elaborate models. A future step in the modelling will be to consider a small (i.e., adhesion area of $50\text{ }\mu\text{m} \times 50\text{ }\mu\text{m}$; thickness $10\text{ }\mu\text{m}$) single cell in close contact (i.e., distance of about 10 nm) with a gate insulator of comparable (or smaller) dimensions. The simulation results will be then utilized for designing real devices, optimized for cellular electrochemical monitoring.

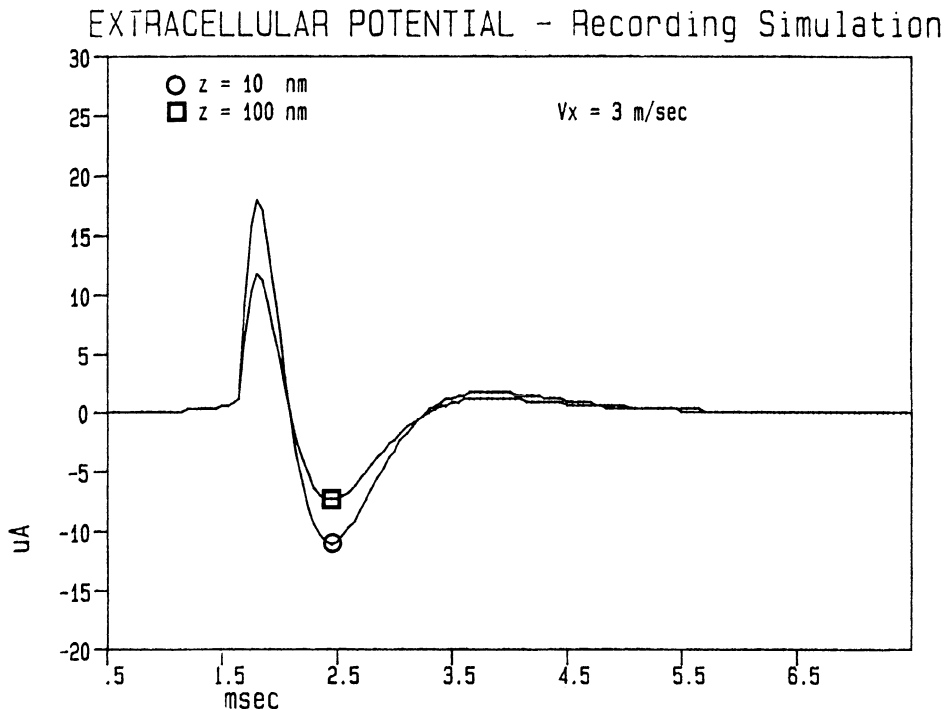


Fig. 4 ISFET drain currents generated by an action potential travelling along a fiber and detected extracellularly. The curves are parametrized by the distance between the recording devices and the fiber.

5. Prospects

Recent advances in microelectronic techniques have made it possible to monitor ,on - line, electrochemical signalling among single units in a network of neurons grown in vitro, in a predetermined pattern. These techniques show great promise in the study of the computational properties of small neural networks reconstructed in vitro. Both micromanufactured planar metallic electrodes and microbiosensors will be very useful in achieving this goal. In our opinion the design of integrated microtransducers, able to detect

chemical species (ions and neurotransmitters) at a cellular or subcellular level, will open a completely new area for computational neuroscience.

Acknowledgment

This work was supported by the National Research Council (CNR) of Italy, Progetto Finalizzato "Materiali e Dispositivi per l'Elettronica a Stato Solido" and by the Italian Ministry of Research.

References

1. A. Grinvald : "Real-time optical mapping of neuronal activity", *Ann. Rev. Neurosci.*, 8 (1985), pp. 263 -281.
2. R. Y. Tsien, M. Poenie : "Fluorescence ratio imaging : a new window into intracellular ionic signaling" *TIBS*, 11, (1986), pp. 450 - 455.
3. J. A. Connor : "Digital imaging of free calcium changes and of spatial gradients in growing processes in single, mammalian central nervous system cells", *Proc. Natl. Acad. Sci.*, 83, (1986), pp. 6179 - 6183.
4. W. G. Wier, M. B. Cannell, J. R. Berlin, E. Marban, W. J. Lederer : "Cellular and subcellular heterogeneity of Ca in single heart cells revealed by Fura-2" *Science*, 235, (1987), pp. 325 - 328.
5. M. Hoppe and J. Beretein-Hahn : "Applications of scanning acoustic microscopy-survey and new aspects", *IEEE Trans. Sonics Ultrasonics*, SU-32, (1985), pp.289 - 301.
6. M. Grattarola, M. Tedesco, A. Cambiaso, G. Perlo, G. Giannetti, A. Sanguineti : "Cell adhesion to silicon substrata : characterization by means of optical and acoustic cytometric techniques" *Biomaterials*, 9, (1988), pp. 101-106
7. J. Hildebrand, D. Rugar, R. Johnston, and C. Quate : "Acoustic microscopy of living cells", *Proc. Natl. Acad. Sci.*, 78, (1981) pp.1656 - 1660.
8. M. Grattarola, A. Cambiaso, S. Cenderelli, and M. Tedesco : "Capacitive measurements in electrolyte-insulator-semiconductor (EIS) systems modified by biological materials" *Sensors and Actuators*, 17, (1989), pp.451 -459.
9. J. A. Hammarback, S. L. Palm, L. T. Furcht and P. C. Letourneau : "Guidance of neurite outgrowth by pathways of substratum-adsorbed laminin", *J. Neurosci. Res.*, 13, (1985), pp. 213 -233.
10. J. A. T. Dow, P. Clark, P. Connolly, A. S. Curtis, and C. Wilkinson : "Novel methods for the guidance and monitoring of single cells and simple networks in culture" *J. Cell Sci.*, Suppl. 8, (1987), pp. 55 - 79.
11. D. A. Robinson, "The electrical properties of metal microelectrodes", *Proc. IEEE*, 56,(1968) pp. 1065 - 1071
12. G. W. Gross, "Simultaneous single unit recording in vitro with a photoetched laser deinsulated gold multimicroelectrode surface", *IEEE Transactions on Biomedical Engineering*, BME - 26, (1979), pp. 273 - 279

13. D. A. Israel, W. H. Barry, D. J. Edell, and R. G. Mark, "An array of microelectrodes to stimulate and record from cardiac cells in culture", *Am. J. Physiol.*, 247, (1984), pp. H669 - H674
14. D. T. Jobling, J. G. Smith, H. V. Wheal, "Active microelectrode array to record from the mammalian central nervous system in vitro", *Med. & Biol. Eng. & Comput.*, 19, (1981), pp. 553 -560.
15. T.F. Freund, G. Buzsaki, O. J. Prohaska, A. Leon and P. Somogyi, "Simultaneous recording of local electrical activity, partial oxygen tension and temperature in the rat hippocampus with a chamber-type microelectrode. Effects of anaesthesia, ischemia and epilepsy" *Neuroscience*, 28, (1989), pp.539 -549.
16. J. W. Parce, J. O. Owicki, K. M. Kercso, G. B. Sigal, H. G. Wada, V. C. Muir, L. J. Bousse, K. L. Ross, B. I. Sikic, H. M. McConnell : "Detection of cell-affecting agents with a silicon biosensor" *Science*, 246, (1989), pp.181 -296.
17. P. Bergveld, J. Wiersma and H. Meertens, "Extracellular potential recording by means of a field effect transistor without gate metal, called OSFET", *IEEE Transactions on Biomedical Engineering*, BME-23, (1976), pp. 136 - 144.
18. W. M. Siu and R. S. C. Cobbold, "Basic properties of the electrolyte-SiO₂-Si system : physical and theoretical aspects", *IEEE Transactions on Electron Devices*, ED-23, (1979), pp. 1805 - 1815.
19. W. G. Regher, J. Pine, and D. B. Rutledge, "A long-term in vitro silicon-based microelectrode-neuron connection", *IEEE Transactions on Biomedical Engineering*, 35, (1988), pp. 1023 - 1032.
20. P. Rosenfalck, "Intra- and extracellular potential fields of active nerve and muscle fibers", *Acta Physiologica Scandinavica*, Suppl. 321, (1969)
21. A. S. Grove, "Physics and technology of semiconductor devices", Wiley, New York, 1967

Part 7

Robot Societies and Self-Organization

A Robot Being

Rodney A. Brooks and Anita M. Flynn
MIT Artificial Intelligence Lab*
Cambridge, MA, *USA*

Abstract

Being a robot in a human habitat requires dealing with cluttered, unconstrained and dynamically changing environments. Most research on autonomous mobile robots assumes a static world. At best, dynamic aspects of the world are to be avoided. We report on a robot, Seymour, which is designed to interact with people while operating in a crowded office environment. Seymour cannot be dynamically told what to do. Rather, like children and dogs, he does what is in his nature (which is determined by programs residing onboard in EPROMs on power up). He pursues his own activities while responding to the presence and actions of nearby people.

Seymour bristles with sensors. But rather than fuse the data from his nine cameras and his pyroelectric array into a world model, he will have many independent perceptual systems which are individually and intimately tied into behavior-generating networks of simple computational elements. Each perceptual subsystem extracts only those aspects of the world which are relevant to the particular task for which it is tuned. Fusion happens closer to the motor level than the sensor level. Seymour uses the modified subsumption architecture which is a methodology for implementing complex agents as an incrementally evolved network of augmented finite state machines.

Our approach in building Seymour and other robots has been inspired in many ways by biological systems and research. In particular, we have adopted an evolutionary method of building complex autonomous agents, where the components are simple distributed computational elements. This gives us strong advantages in dealing with the complexity of the environment. We are not particularly interested however, in simply reproducing the complexity of Nature's solutions. In fact, we maintain that biological inspirations can be taken too far. In particular introspection to determine how perception or even reasoning works is bound to fail.

1 Introduction

Our aim is to build robot beings that exist in the world of human beings, operating as autonomous agents. Such a being is not told what to do by an operator. Rather it is switched on and does what is in its ROM-determined nature to do given the circumstances in which it finds itself.

*Support for this research was provided in part by the University Research Initiative under Office of Naval Research contract N00014-86-K-0685, in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014-85-K-0124 and in part by a gift from Siemens.

To illustrate this distinction we can consider two robots whose task is to deliver processed metal parts from one set of machines in a factory to another set.

One robot is connected to a central Computer Integrated Manufacturing computer. The central computer monitors the output of each machine and plans collection runs for the robot to go by certain machines, pick up a certain number of parts, and then to go to the delivery station, or stations, and deposit them. The central computer downloads a set of commands to the robot which then executes them, reporting back for instructions when something goes awry, for example when an obstacle blocks the pre-planned path. At set times the central computer commands the robot to drive to a recharging station.

The Robot Being, on the other hand, is completely autonomous. It actively looks for machines of the sort that produce the appropriate sorts of metal parts. It may do this through general purpose vision, or it may be that all such machines have a certain bar code attached to them, so that suffices for recognition. If new such machines are installed, the robot being will likely locate them without having to be explicitly informed of their existence. The robot being picks up finished parts and makes its way to where they should be delivered, autonomously solving any problems that arise along the way. For certain classes of blocked paths it finds a way around the obstacles. For more complex situations it might simply stop, and wait for the path to be cleared. After all it is not a super being—just a simple robot being. It monitors its battery level and trades off its need to be recharged against how many of its brethren it observes carrying out the same task. At a certain point it goes off to the recharging station on its own initiative.

Clearly it is possible to construct useful robots in between these two extremes of autonomy (although the first example is often considered very autonomous by today's standards). In this paper we want to concentrate on robots inclined towards the second extreme. As the environment becomes more unstructured, e.g., in the transition from the factory, to the home, to agricultural work, to military situations, to planetary exploration, robots of the latter type will become more useful, and will be clearly superior to those of the first type.

Let us be more clear about what we mean by a *Robot Being* by defining the term:

A Robot Being is robot which "lives" in the world, carrying out its own agenda of on-going projects, while maintaining the necessary balance with its environment to ensure its continued successful operation.

Notice that the term "robot being" entails a point of view of the designer of such a robot. It is no longer an I/O device, but rather an active participant in the world.

We maintain that robot beings cannot be built readily by decomposing their intelligence into distinct perceptual, modeling, reasoning, planning, and execution modules. Such a decomposition into distinct information processing modules creates robots that take a snapshot from their sensors, reconstruct a complete model of the world, then decide how to act. We argue instead that robot beings must be based on the guiding principle that the robot is *situated* in the world—i.e., it is not a separate system from the world, but rather a component of the world with which it must interact, and be interacted with by, all the other components of the world. It has an on-going purpose in its existence, albeit one built in by the human engineers who constructed it, and must rely on whatever in-built resources it has, to maintain that purpose in an unstructured world.

A robot being is a large collection of triggerable tight sensing-action servo loops running in parallel. It is not a collection of hierarchically controlled subroutines.

[Moravec 84] argued that mobility is the key driver in the development of intelligence. In [Brooks 87] we argued that the world is its own best model, and that traditional Artificial Intelligence notions of world modeling are both impractical and unnecessary in building robot beings. [Agre and Chapman 87] have also been strong proponents of this point of view. We have developed a number of robots based on these principles [Brooks 86], [Brooks and Connell 86], [Connell 87], [Horswill and Brooks 88], [Brooks, Connell and Ning 88], [Flynn and Brooks 88], [Brooks 89], [Connell 89] and [Flynn, Brooks, Wells and Barrett 89].

Our approach bears many similarities to that proposed by [Minsky 86] for explaining how the brain works. Both approaches rely on a collection of almost independent agents interacting in such a way that globally consistent behavior emerges. The advantages of this approach are two-fold in that they help to solve two difficult problems for mobile robots.

The changing world problem. To deal with a changing world the multiple agents operate in parallel, ensuring that the robot is able to react in a timely fashion to ongoing changes in the environment.

The sensor fusion problem. With independent agents, the problem of fusing multiple sensory inputs into a consistent world model disappears, as each agent need only extract the aspects of the world that bear on its own operation.

We have built a large number of robots which demonstrate aspects of this approach. We are designing, building and shaking down a new robot, Seymour, that eventually will extend the level of autonomy and self sufficiency previously demonstrated. We have previously reported on certain unique hardware aspects of this robot [Ciholas 88] and on special purpose perceptual systems developed in support of the robot [Brooks, Flynn and Marill 87], [Viggh and Flynn 88], [Sarachik 89]. In this paper we contrast Seymour with previous robots and report on a new language in which he is programmed.

2 Other Robots

Before embarking on our description of Seymour and his intelligence, we digress to contrast the robot being approach with previous work on mobile robots.

2.1 Prehistory

One of the earliest mobile robots was that produced by [Walter 50], [Walter 51]. It was a small motorized “turtle” with a “brain” built from two vacuum tubes. It predated the field of Artificial Intelligence and so was free from the later prejudices on the structure of cognition that AI has promulgated for the last 30 years. Walter’s machine(s) had two simple reflex actions stimulated by light, essentially *seek* and *flee*, but whose selection was modulated by its own battery state. It was able to demonstrate a surprisingly rich set of behaviors, many of which had obvious interpretations as analogs to animal behaviors.

According to our definition, this was a robot being as it carried out its business in a dynamic world, without receiving detailed instructions from a human.

Inspired by an illustrious career in neuroscience [Braitenberg 84] has continued Walter's approach in a series of thought experiments to constructing simple vehicles that are sometimes attracted and sometimes repelled by lights. The first few experiments are quite believable and enlightening, but unfortunately in later experiments which are more complex, but yet receive less detailed descriptions, he ignores serious issues concerning the sensor fusion problem.

2.2 The Grand Planners

A number of mobile robots have been built which rely on building and/or maintaining detailed world models, and then planning a series of actions within those world models. In our view, none of these robots qualify as robot beings.

In the late sixties, an ambitious mobile robot project was undertaken at the Stanford Research Institute (now SRI International) which has had major impacts on the direction of both Artificial Intelligence and mobile robot research through today. The robot was named "Shakey" and a collection of reprints of many of the technical reports written at the time can be found in [Nilsson 84].

Shakey had a complete predicate calculus based world model of the artificial and carefully engineered environment in which it was placed. That environment consisted of a number of specially built rooms, with carefully painted walls and dark black baseboards, along with a set of large "toy" blocks a few feet on a side painted in such a way that each face was a single matte color. Odometry was used to update the position estimate of the robot, but since that was subject to drift, a simple visual calibration technique was used. Using a prediction of the appearance of an interior trihedral vertex (e.g. where the two walls and the floor met) Shakey could update his position estimate by actually locating it (not too hard given the highly contrasting baseboards).

Shakey was given high level commands in pseudo English over a teletype. Typical commands might be to go to a particular room, or block a particular doorway (this could be achieved by pushing one of the large toy blocks so that it fit in the doorway). It was then up to Shakey to plan (using a planning system called STRIPS developed especially for Shakey, which shaped the next 20 years of planning research in AI) a series of actions to carry out the commanded task. Once the task was completed (perhaps involving dynamic replanning in response to simple perceptual events, such as blocked paths), Shakey waited for another English command. Shakey then, is not a robot being in our sense. He did not have on-going autonomous projects which defined his existence.

During the mid-seventies and continuing through to today, the robot Hilare [Giralt, Chatila and Vaisset 84] was developed at LAAS in Toulouse. Early in the project's life the emphasis with this robot was on building accurate two dimensional maps of its environment which consisted of an open space partitioned with moveable (by the operators) vertical plywood walls. These maps were to be used later for planning to carry out higher level tasks commanded by human operators. However while building a map, Hilare did show certain characteristics of a robot being. It had an ongoing project to build a map, and would autonomously decide where to travel to next, in order to get the best view of the unknown parts of the world. While doing that it would plan out collision free

paths for itself and generally operate in a totally autonomous manner. It did not have to receive any special purpose commands in order to map out any particular environment. Its general purpose mapping program was meant to handle them all. Unfortunately, it could only handle a static environment.

More recently, the emphasis with Hilare has shifted [Noreils and Chatila 89] towards dealing more directly with the two problems identified earlier; the dynamic world and sensor fusion problems. They have defined an interesting high level language for specifying "missions" for the mobile robot. Within a mission the robot acts as a robot being, making heavy use of traditional planning methods, but the top level view is still that a human downloads a mission to the robot which the robot then carries out.

[Lin, Mitchell, Philips and Simmons 89] have built a robot using classical AI planning techniques which can carry out a variation of a soda can collecting task developed using the robot being approach [Connell 89]. The robot lives in a single room with a camera mounted in the ceiling. As the camera detects a large cup-like object, a planner directs the robot to go towards it, localize it with hand mounted sonars and collect it. This robot does have an on-going project which it pursues autonomously, but it is restricted to a specially engineered environment with an overhead camera.

The Hermes robot [Mann, Hamel and Weisbin 88] operates in a very carefully constructed environment (bubble wrap covers all surfaces to make sure that they are visible to sonar) using classical planning techniques. This robot is very much in the mission oriented tradition with specific sequence commands being downloaded for a particular task.

The Yamabico series of robots have been used for a number of component demonstrations. Recently much work by [Kanayama and Miyake 86] has been directed at enabling the robot to precisely follow a predefined trajectory. The desire for such performance is deeply rooted in the tradition of having a high level planner hierarchically determine a sequence of actions for a robot, and sending them down to lower level execution modules which will blindly carry them out no matter what the cost.

2.3 The Reality of Perception

Much good work has been done in the field of mobile robotics where the robots have not even reached the stage of being downloaded particular missions, let alone been let loose as robot beings. Rather the work has concentrated on the very difficult problems of perception in unstructured environments based on a mobile platform. We very briefly describe some of the work of this nature that has been done on actual mobile robots.

Using two robots, the Cart and Neptune, [Moravec 83] demonstrated visual navigation in static but cluttered real world spaces. He achieved this at a time when almost no object recognition programs worked except in the most contrived of circumstances. The reason for his success was that he simply tried to distinguish where there was material, and where there was not material and made the robots navigate where no material had been detected.

[Kriegman and Binford 88] demonstrated visually guided corridor following and door finding on the robot named Mobi. It used a linear camera stereo algorithm for depth perception. Mobi did not really have a project to participate in, just a straightforward

mission to wander down a corridor. The corridor environment was rather benign, with controlled lighting, uniform surfaces and no dynamic aspects.

[Horswill and Brooks 88] used the robot Allen to demonstrate a number of simple visually guided behaviors. A subsumption-like network enabled the robot to track and follow moving objects. One behavior looked for motion in order to select a target, and another compared successive images in order to track it. The following behavior tried to servo the robot to keep the object in a fixed location on its uncalibrated camera image. Internally, the network often switched between behaviors, but this was invisible to an external behavior—it just seemed as if Allen was pursuing the target. Additions to the basic network let Allen select small static targets to approach, and also let him follow corridors, by trying to follow the piece of floor just a little way in front of him—it kept moving as he did.

Using the robot Tito, [Sarachik 89] demonstrated two visually guided behaviors which will be used in support of a robot being. Each used a stereo pair of linear cameras. A vertically mounted pair made use of rotational motions of the base to produce images from which the dimensions of the room could be extracted even though the camera system was uncalibrated. Then employing earlier results from [Brooks, Flynn and Marill 87] the robot used forward motion to calibrate a horizontally mounted pair of cameras, which were used to find doorways through which the robot drove. The continuing scenario then called for the robot to topologically map the newly entered room and gradually build a complete map of the world. There were no particular restrictions on the amount of clutter allowed in the rooms (they were typical laboratory and office rooms) nor on the presence of people and other dynamic moving shapes. This work is a clear precursor to a robot which includes amongst its activities or projects, the task of mapping the world.

The ALV (Autonomous Land Vehicle) sponsored by DARPA was initially intended to operate in a manner similar to that which we have described for robot beings. It was planned to demonstrate autonomous planning and navigation over road networks and rough off-road countryside. The actual demonstrations reported [Turk, Morgenthaler, Gremban and Marra 88] however, turned out to be more like tests of vision systems in fairly tightly constrained circumstances. The vehicle drove along a paved road and using an adaptive histogram segmentation technique was able to find and track the road from image to image and steer towards its. The ALV is a large custom-built vehicle weighing approximately 12,000 pounds.

A similar result, although using a Hough transform technique based on possible road geometry rather than a histogram of intensities, was demonstrated by [Thorpe, Hebert, Kanade and Shafer 88] on their robot named Navlab. Navlab is a modified delivery van.

[Flynn 88] reported early experiments with ROBART II (see below), where sonar and active infrared intensity ranging were used to complement each other and build more accurate maps than could be done with either sensor alone. Sonar provided good range but poor angular resolution, while active infrared provided good angular resolution, but poor range. Together they provided a map with good enough range and angular resolution that traditional path planners could be used on them.

2.4 Autonomous Agents

Recently there has been some work on mobile robots which just about qualify as robot beings, given our definitions. All of these robots have onboard computation.

Herbert [Connell 89] runs the subsumption architecture [Brooks 86]. It is a completely autonomous robot with 24 onboard processors, communicating over low bandwidth (240 baud) serial lines. It has 30 infrared proximity sensors and a laser light stripper which delivers a 256×32 pixel depth map once every second. It also has an onboard compass which delivers about 3 bits of orientation information. It has a two degree of freedom manipulator with a parallel jaw gripper, capable of reaching down to ground level and high enough to pick things up off an ordinary desk or bench. The gripper, or hand, has a number of sensors mounted on it, including a novel pair of crossed active infrared sensors, and various touch switches. The robot's task is to operate in an unknown environment, look for soda cans, pick them up, potentially from a cluttered desk, and return them to the location at which it was switched on. An interesting aspect of this particular robot is that no state lingers internally in the robot for more than 3 seconds. The way it knows whether it should be exploring or returning home is by whether its hand is grasping a soda can. The individual behaviors have been demonstrated hundreds of times and are well documented. The complete behavior of the robot was demonstrated in mid-June 1989 as a finale for the author's Ph.D. thesis. If Herbert's mean time between catastrophic hardware failure were longer than 15 minutes it would probably qualify as a robot being.

Genghis [Brooks 89], [Angle 89] is a six legged walking robot that weighs one kilogram. It has four onboard processors, one of which runs the subsumption architecture, twelve motors, twelve force sensors, six pyro-electric sensors, two whiskers and two inclinometers. Genghis can scramble over rough terrain. It stands motionless waiting for an infrared emitting body to come into its field of view whereupon it starts crawling towards its newly acquired target. It is relentless in its pursuit and not stopped by intervening obstacles (hence its name). Its project is to follow moving bodies.

Flakey [Kaelbling 86] is a successor to Shakey at SRI International. It has sonar and a TV camera (and a 40Mb hard disk!). Vision has been used for corridor tracking [Wells 87]. Sonar is used for wall following and doorway identification. A novel architecture [Kaelbling 88] is used to compile a goal language into a gate level circuit, and then into a circuit simulator in 68000 machine language. The robot has been demonstrated navigating in an office world, with and without the help of a map.

Harvey and George [Arkin 87] are two instances of Denning Research Robots (see below) that have been driven using a Schema architecture. The primary method of goal fusion on these robots is through addition of potential fields, where each field is specific to a particular goal. Recent work [Arkin 89] has been aimed at monitoring the internal functioning of the robot (such as battery level, rate of battery drain, temperature), making it more self sufficient. The robots have been demonstrated navigating both indoors and outdoors in cluttered dynamic environments.

The two robots that come closest to being robot beings are meant to operate as autonomous guards or sentries. Such robots really do have an ongoing project; detect intruders and make a report. To be cost effective they must be completely autonomous. To be effective guards they must be tamper proof, so that intruders can't switch them off before a warning is given. Being heavy is another advantage too, as sometimes the

items being guarded by a robot are worth much less than the robot itself, leading to the situation where it might be most profitable to steal the guard!

Robart II [Everett and Gilbreath 89] is an experimental sentry robot. It has a number of intruder sensors, and uses sonar as its primary map building and navigation sensor. A novel feature of Robart is its very complete self diagnostic system, which ranges from checking that all cables are correctly in place, to run time monitoring of its own behavior in order to check for hardware failures.

Denning Mobile Robotics, Inc., has been developing a large sentry robot for many years. Many research versions of the vehicle have been used by research labs all over the world. They have not published papers, but they have built a number of sentry prototypes that have been temporarily installed as security robots in various buildings. One at the Massachusetts Exposition Center ran for 30 nights with automatic recharge cycles during the day without any human intervention. The vehicles use a combination of sonar and infrared beacons for navigation.

3 The New Subsumption

Over the last four years we have been developing and refining the subsumption architecture [Brooks 86], [Brooks 89], [Connell 89]. In the past we have always concentrated on the composition of simple behaviors constructed from augmented finite state machines. Indeed networks of augmented finite state machines are still our primary computational model at one level. But recently we have developed a new formalism for describing behaviors partially inspired by the work of [Maes 89]. We call this new formalism the *Behavior Language*. It is still under active development, but we report on progress with it for the first time here. Our current approach is to compile the behavior language into our traditional networks of finite state machines (with some minor extensions), in order to make use of the multiple backends we have developed for that language running on different machines (e.g., the Hitachi 6301, the Motorola 6811, and the Motorola 68000).

3.1 Essential Subsumption

In this section we briefly recap the subsumption architecture as it has developed over the last few years. Details can be found in [Brooks 86] and [Brooks 89].

The key idea behind the subsumption architecture is that a given level of performance should be augmentable by adding more computational processes. A subsumption program is arranged in an evolutionary set of layers. It need not be a linear set of layers; figure 1 shows a network of layers used in building the subsumption program to control a six legged walking robot [Brooks 89]. The arrows in that diagram go from earlier layers to later layers. Each layer connects sensors to actuators as in figure 2. A later layer does not call earlier layers as a subroutine, but the designer of a later layer can assume that the lower level layer will be there and be operating. Thus for instance, a layer which is designed to give a robot the ability to go down a corridor by following the walls does not have to deal with avoiding collisions along the way if there is a lower level layer that is designed to guarantee that collisions will not occur.

Each layer is constructed from a network of augmented finite state machines.

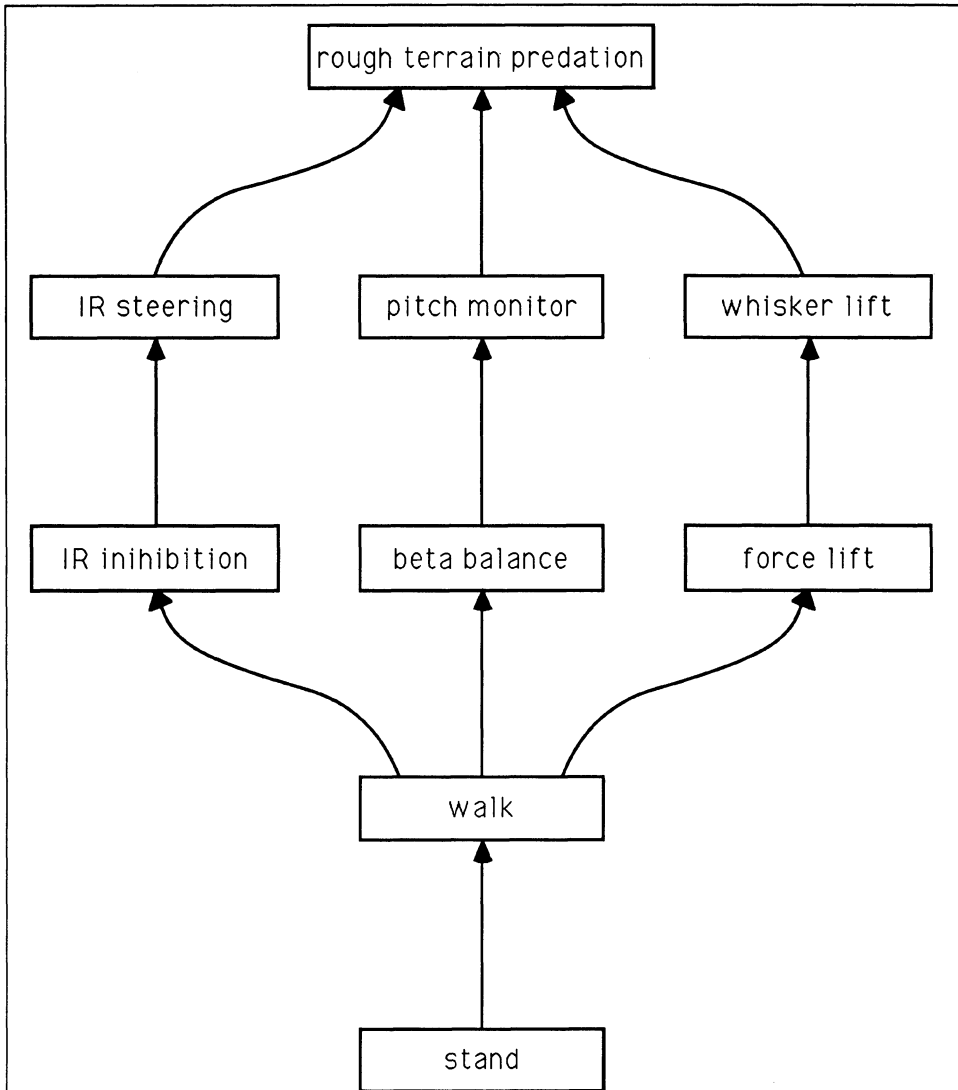
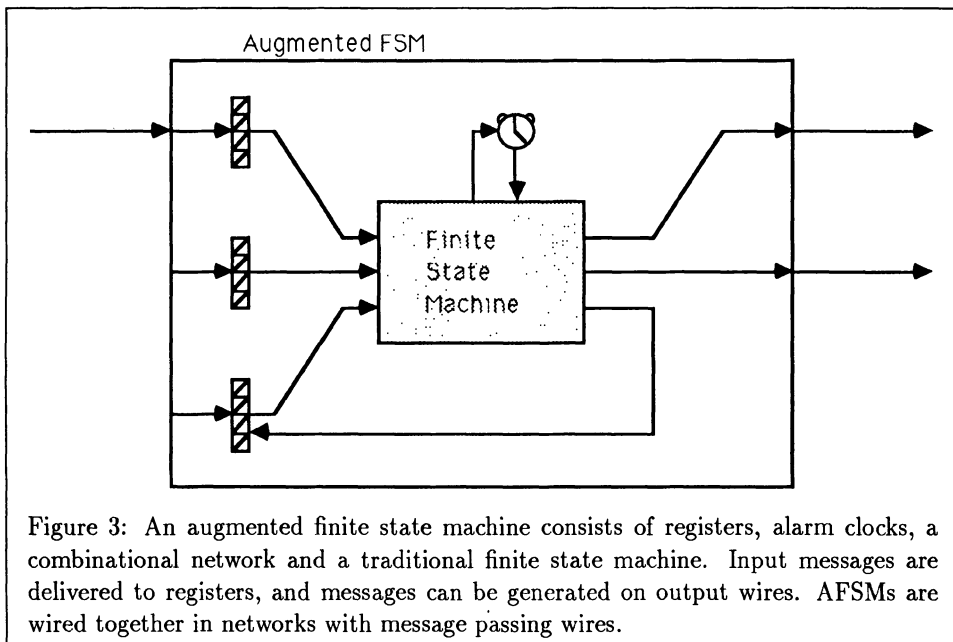
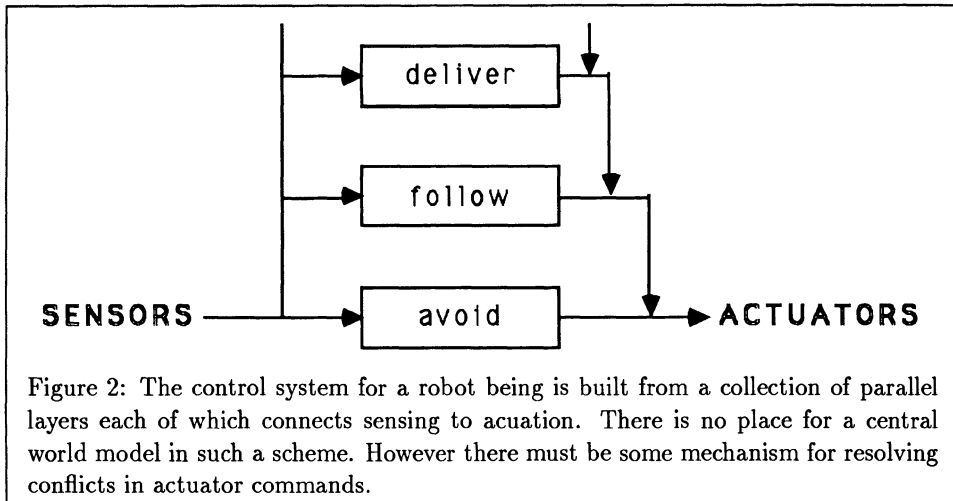


Figure 1: These are the conceptual layers used to program the six legged walking robot described in [Brooks 89]. The vertically lowest layers were built first, and later layers assume the existence of those layers, although they do not invoke them as subroutines. Rather they coexist as parallel behaviors.



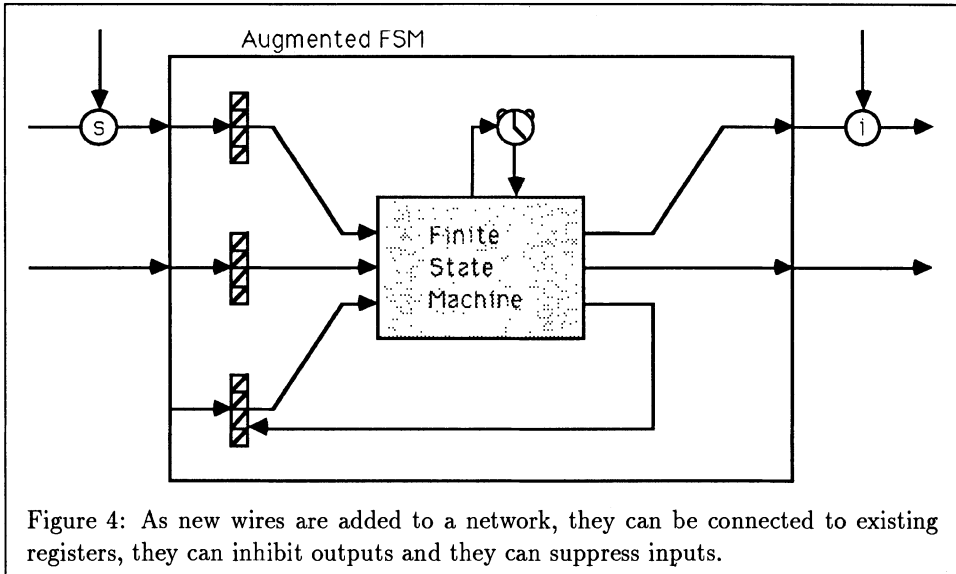


Figure 4: As new wires are added to a network, they can be connected to existing registers, they can inhibit outputs and they can suppress inputs.

Each augmented finite state machine (AFSM), figure 3, has a set of registers and a set of timers, or alarm clocks, connected to a conventional finite state machine which can control a combinational network fed by the registers. Registers can be written by attaching input wires to them and sending messages from other machines. The messages get written into the registers by replacing any existing contents. The arrival of a message, or the expiration of a timer, can trigger a change of state in the interior finite state machine. Finite state machine states can either wait on some event, conditionally dispatch to one of two other states based on some combinational predicate on the registers, or compute a combinational function of the registers directing the result either back to one of the registers or to an output of the augmented finite state machine. Some AFSMs connect directly to robot hardware. Sensors deposit their values to certain registers, and certain outputs direct commands to actuators.

A series of layers of such machines can be augmented by adding new machines and connecting them into the existing network in the ways shown in figure 4. New inputs can be connected to existing registers, which might previously have contained a constant. New machines can inhibit existing outputs or suppress existing inputs, by being attached as side-taps to existing wires (figure 4, circled 'i'). When a message arrives on an inhibitory side-tap no messages can travel along the existing wire for some short time period. To maintain inhibition there must be a continuous flow of messages along the new wire. (In previous versions of the subsumption architecture [Brooks 86] explicit, long, time periods had to be specified for inhibition or suppression with single shot messages. Recent work has suggested this better approach [Connell 88].) When a message arrives on a suppressing side-tap (figure 4, circled 's'), again no messages are allowed to flow from the original source for some small time period, but now the suppressing message is gated through and it masquerades as having come from the original source. Again, a continuous supply of suppressing messages is required to maintain control of a side-tapped wire. One

last mechanism for merging two wires is called defaulting (indicated in wiring diagrams by a circled 'd'). This is just like the suppression case, except that the original wire, rather than the new side-tapping wire, is able to wrest control of messages sent to the destination.

Inhibition and suppression are the mechanisms by which conflict resolution between actuator commands from different layers (see figure 2) is achieved. Notice that in this definition of the subsumption architecture, AFSMs can not share any state, and in particular they each completely encapsulate their own registers and alarm clocks.

All clocks in a subsumption system have approximately the same tick period (0.04 seconds in most of our robots), but neither they nor messages are synchronous. The fastest possible rate of sending messages along a wire is one per clock tick. The time periods used for both inhibition and suppression are two clock ticks. Thus, a side-tapping wire with messages being sent at the maximum rate can maintain control of its host wire. We call this rate the *characteristic frequency* of the particular subsumption implementation.

3.2 The Behavior Language

Our new behavior language groups multiple processes (each of which usually turns out to be implemented as a single AFSM) into *behaviors*. There can be message passing, suppression, and inhibition between processes within a behavior, and there can be message passing, suppression and inhibition between behaviors. Behaviors however act as abstraction barriers, and one behavior cannot reach inside another.

Within a behavior each process is much like an AFSM and indeed our compiler for the behavior language converts them to AFSMs. However they are generalized so that they can share registers. Furthermore there is a slightly more general mechanism than the original alarm clocks, namely monostables. Monostables too can be shared between processes within a single behavior.

A behavior is specified by describing a collection of processes. The building blocks for specifying a process are condition-action rules, called **whenever** clauses. A **whenever** clause specifies that when a certain condition is true, then a sequence of actions should be performed. Some of those actions might themselves be **whenever** clauses. The computational model for a **whenever** clause is that once entered, it waits for its condition to be true, then evaluates the sequence of actions, then waits for the condition to be true again. Once entered, the only possible way to exit a **whenever** clause is via an explicit non-local exit called **done-whenever**; it only makes sense to exit a **whenever** clause which is not at the top level.

Examples of **whenever** clauses follow:

- a. (**whenever** (received? mess)
 (output out (+ 3 mess)))
- b. (**whenever** (delay 0.5)
 (ping-sonars))
- c. (**whenever** (< (infrared 3) (meters 1.5))
 (output velocity (meters 1.0))

```
(delay 0.5)
(output halt 't))
```

The first example (a) waits until a message is received on the `mess` input port, then sends it along to the `out` port incremented by 3. Example (b) ensures that the sonars get pinged every half a second. In example (c) the test is carried out at the characteristic frequency of the particular implementation. Whenever it is true, two messages are output, with a half second delay between them. Immediately the condition is tested again, which in this case might lead to buggy behavior.

Another **whenever** can be embedded within a **whenever** clause body. In this way a process can have multiple wait states.

Multiple **whenever** clauses can be made the body of an **exclusive** clause. The semantics of an **exclusive** clause are that all of the subordinate clauses are waiting on their conditions to be true, and when one becomes true that **whenever** clause body is entered and the remaining **whenever** clauses are abandoned. Upon completion of the chosen body the toplevel **exclusive** clause is entered again, and all the conditions are again waited upon in parallel.

A detailed example of a behavior can be seen in Appendix A.

Behaviors are wired together via **connect** statements. The compiler takes care of ensuring that all necessary internal connections are made from a single external connection.

4 Seymour the Robot

Our latest robot is named Seymour and is shown in figure 5. This robot is the focal point of many lines of research over the last four years, and we plan on demonstrating high levels of completely autonomous behavior. Seymour should be a true robot being.

4.1 Mechanical Base

The base is one co-developed by our laboratory and a small local company, Real World Interface. It is three-wheeled, 30cm in diameter, and uses a synchro-drive. It has one motor to steer all three wheels together and one to drive all three wheels together. Its top speed is 2.2 meters per second, or about 8 kilometers per hour.

The base includes a servo computer, an NEC motor controller chip, which can communicate with Seymour's main host computers via a 9600 baud serial port. The servo computer runs PID algorithms to drive the motors to set positions at set velocities, with set acceleration profiles. At any time, any such parameter can be changed and the controller responds appropriately. Commands are sent from the host computer as two byte ASCII sequences followed by numeric arguments. Additional commands poll the base on battery status and motor currents. The host computer can thus check for collisions as shown in actual code in Appendix A. Higher velocity control is attained in that code by continuously sending down new and far off position targets for the servos.

The high velocities of which Seymour is physically capable are a challenge to sensor and perception capabilities.

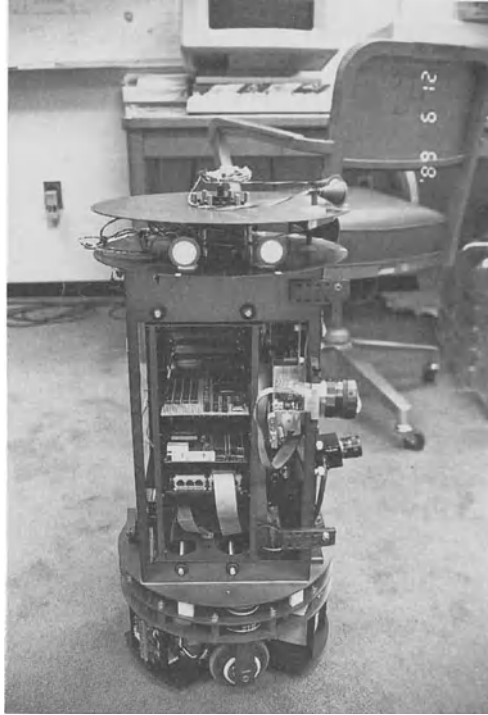


Figure 5: Seymour the robot has nine onboard cameras, a rotating head of pyroelectric sensors, and room for four onboard 68000s and two digital signal processors.

4.2 Sensor System

Seymour is an all passive-sensing robot. Active sensors, such as laser rangefinders, sonars or infrared rangers, emit radiation and measure the return signal (usually a modulated signal so that the signal to noise ratio can be very high). Passive sensors emit no radiation, but instead rely on the ambient radiation that happens to fall on them accidentally. The interpretation of raw data is much harder with passive sensors.

Most animals rely on passive sensors. It seems the major reason might be that energy budgets preclude active sensing from all but a narrow zone around the organism. Once perception has developed to make use of passive information acquired from far off sources it can handle the close in zones relatively more easily. Thus passive sensing appears to be an interesting constraint within which to do perception. If it leads to a very high performance system, the payoffs will be immense.

Seymour has nine CCD cameras on board. Six of them are custom built linear cameras [Ciholas 88], and three are standard NTSC cameras. In all cases, we subsample down to about 1000 pixels per camera. The cameras run at about 10 frames per second, giving a total of only 90,000 pixels per second that need be processed. The linear cameras can actually be run at 140 frames per second, but just one camera at that rate would have a higher data rate than we currently plan for the whole robot. The linear cameras do have program controllable exposure times, so one very simple subsumption behavior is to keep the cameras auto-exposed as Seymour wanders into brighter or darker environments.

Two forward looking linear cameras will be used for obstacle avoidance [Brooks, Flynn and Marill 87] and door finding [Sarachik 89]. Two sideways looking linear cameras will be used for wall and doorway finding [Bolles, Baker and Marimont 87]. Two upward looking linear cameras will be used for room mapping [Sarachik 89]. The forward looking square cameras will be used for corridor following, and interesting object finding [Horswill and Brooks 88], and potentially for people localization from a short range.

The other major sensor we use is passive infrared. Infrared detectors come in two varieties. The first is similar to the silicon-based CCD sensor used in a normal camera, except that it is designed to be sensitive only to the long-wavelength radiation in the infrared portion of the spectrum. The problem with these “night vision” cameras is that they normally have to be cooled, forcing the robot to carry its own supply of liquid nitrogen. A second type of passive infrared detector, one that we use, is the pyroelectric sensor. These are room temperature devices but unlike night vision cameras, they don’t respond to steady state infrared radiation. Pyroelectrics respond only to a *change* in temperature across their field of view. This is due to the use of special materials which have a crystalline structure that cause dipoles to align when exposed to a temperature change. The movement of these charge dipoles creates a current which can be detected. The current returns to zero when the temperature reaches a steady state. Consequently these pyroelectric sensors can be viewed as *warm-body motion sensors*. Typically, infrared filtering windows are placed over the sensing elements so that they respond only to wavelengths emitted by humans. Widespread use of these sensors can be found in burglar alarms and security systems.

The pyroelectric sensor suite on Seymour is used to enhance people location. Normally, pyroelectric sensors are single-pixel point sensors and only give an indication of whether

or not a person has moved through the field of view. However, by placing several sensors around the perimeter, Seymour can obtain a 360 degree coverage. Furthermore through a clever geometric trick using rotating sensors [Viggh and Flynn 88] it is possible to determine both heading and range to a target.

Due to the fact that pyroelectrics respond to a change in temperature over time, there always needs to be relative motion between the sensor and the person. There are three possible sources of relative motion: a moving robot containing a fixed sensor, a still robot holding a movable sensor, or a moving person. For the case of a still person and an unmoving robot, we can acquire heading information to the person by utilizing two sensors that scan. More complete coverage can be obtained by placing several sets of sensors around the perimeter of the robot. The case of a moving person is handled with fixed sensors by observing the sequence of sensor firings as the person passes through subsequent sensors' fields of view. Essentially then, fixed sensors can detect moving people, and moving sensors can detect still people, and we know to switch between modes by noting when sensor readings go out of range.

4.3 Computational Hardware

Seymour has a card cage with 14 slots organized into four backplanes 5, 2, 5, and 2 slots each. Each backplane is mastered by a custom CMOS 68000 board, that includes two serial ports, a parallel port, a network port and 192Kb of RAM. The 68000s are interconnected via a high speed network (not yet operational)

There are a number of additional boards which can be plugged into these buses. One provides a 256Kb expansion along with six additional serial ports. It also supports a 128×160 bit map Toshiba LCD display. Two such displays are mounted on Seymour. Another board handles the linear cameras, grabbing images from up to eight such cameras. Yet another board (under development) does frame grabbing from the two dimensional NTSC cameras.

Low level vision is done on digital signal processing boards on the backplanes of two of the 68000s. These are based on the Analog Devices ADSP-2100 chip, a 16 bit integer digital signal processor. We have a compiler [Horswill 89] for a vectorized Scheme-like language for these processors. Early experiments with the linear obstacle avoidance algorithms of [Brooks, Flynn and Marill 87] indicate very significant speedups with these processors.

Additional boards are under development for such tasks as sound understanding and speech generation.

5 Seymour the Being

Seymour, being a complicated robot, has only just become operational (May 1989) after almost two years of developing hardware and software infrastructure to support the robot.

At this writing (June 1989) Seymour has operated in a blind mode, bumping around the world and bouncing off obstacles using current overload sensing. The pyroelectric sensors have sent valid range data into the subsumption architecture, and the linear cameras have fed data to the digital signal processors.

We have successfully demonstrated the new behavior programming language on Seymour. In fact, all the programs that run on Seymour are written in this rule-like language. The behavior language hides much of the complexity that was inherent in the earlier AFSM model.

Seymour has a richer sensor suite than any of our previous robots. Furthermore, the computational hardware onboard Seymour can support approximately 4000 AFSMs. None of our previous robots have used more than 100 AFSMs. The behavior language is a necessary piece of infrastructure for Seymour to meet his full potential.

6 References

- [Agre and Chapman 87] "Pengi: An Implementation of a Theory of Activity", Philip Agre and David Chapman, *AAAI-87*, Seattle, WA, July, 268-272.
- [Angle 89] "Genghis, a Six Legged Autonomous Walking Robot", Colin M. Angle, *MIT S.B. Thesis in Electrical Engineering and Computer Science*, March.
- [Arkin 87] "Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior", Ronald C. Arkin, *Proceedings IEEE Robotics and Automation*, Raleigh, NC, March, 264-271.
- [Arkin 89] "Dynamic Replanning for a Mobile Robot Based on Internal Sensing", Ronald C. Arkin, *Proceedings IEEE Robotics and Automation*, Scottsdale, AZ, May, 1416-1421.
- [Braitenberg 84] "Vehicles: Experiments in Synthetic Psychology", Valentino Braitenberg, *MIT Press*, Cambridge, MA.
- [Bolles, Baker and Marimont 87] "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion", Robert C. Bolles, H. Harlyn Baker and David H. Marimont, *International Journal of Computer Vision*, 1:1, 7-55.
- [Brooks 86] "A Robust Layered Control System for a Mobile Robot", Rodney A. Brooks, *IEEE Journal of Robotics and Automation*, RA-2, April, 14-23.
- [Brooks 87] "Intelligence Without Representation", Rodney A. Brooks, *Preprints of the Workshop on Foundations of Intelligence*, MIT, Endicott House, June.
- [Brooks 89] "A Robot that Walks: Emergent Behavior from a Carefully Evolved Network", Rodney A. Brooks, *Neural Computation* 1:2, Summer.
- [Brooks and Connell 86] "Asynchronous Distributed Control System for a Mobile Robot", Rodney A. Brooks and Jonathan H. Connell, *SPIE Vol. 727 Mobile Robots*, Cambridge, MA, November, 77-84.
- [Brooks, Flynn and Marill 87] "Self Calibration of Motion and Stereo for Mobile

Robot Navigation”, Rodney A. Brooks, Anita M. Flynn, and Thomas Marill, *MIT AI Memo 984*, August.

[**Brooks, Connell, and Ning 88**] “Herbert: A Second Generation Mobile Robot”, Rodney A. Brooks, Jonathan H. Connell, and Peter Ning, *MIT AI Memo 1016*, January.

[**Ciholas 88**] “A Dual-Processor Vision System for Real-Time Stereo and Motion”, Michel E. Ciholas, *MIT S.M. Thesis in Electrical Engineering and Computer Science*, June.

[**Connell 87**] “Creature Building with the Subsumption Architecture”, Jonathan H. Connell, *IJCAI-87*, Milan, August, 1124–1126.

[**Connell 88**] “A Behavior-Based Arm Controller”, Jonathan H. Connell, *MIT AI Memo 1025*, June.

[**Connell 89**] “A Colony Architecture for an Artificial Creature”, Jonathan H. Connell, *MIT Ph.D. Thesis in Electrical Engineering and Computer Science*, June.

[**Everett and Gilbreath 89**] “ROBART II: A Robotic Security Testbed”, H. R. Everett and G. A. Gilbreath, *Naval Ocean Systems Center Technical Document 1450*, San Diego, CA, January.

[**Flynn 88**] “Combining Sonar and Infrared Sensors for Mobile Robot Navigation”, Anita M. Flynn, *Robotics Research 7:6*, 5-14.

[**Flynn and Brooks 88**] “MIT Mobile Robots: What’s Next?”, Anita M. Flynn and Rodney A. Brooks, *Proceedings IEEE Robotics and Automation*, Philadelphia, PA, April, 611–617.

[**Flynn, Brooks, Wells and Barrett 89**] “The World’s Largest One Cubic Inch Robot”, Anita M. Flynn, Rodney A. Brooks, William M. Wells and David S. Barrett, *Proceedings IEEE Micro Electro Mechanical Systems*, Salt Lake City, Utah, February, 98–101.

[**Giralt, Chatila and Vaisset 84**] “An Integrated Navigation and Motion Control System for Multisensory Robots”, Georges Giralt, Raja Chatila and Marc Vaisset, *Robotics Research 1*, Brady and Paul eds, MIT Press, 191–214.

[**Horswill 89**] “The SenseLisp Programmer’s Manual”, Ian D. Horswill, *MIT AI internal memo*, May.

[**Horswill and Brooks 88**] “Situated Vision in a Dynamic World: Chasing Objects”, Ian D. Horswill and Rodney A. Brooks, *AAAI-88*, St Paul, MN, August, 796–800.

[**Kaelbling 86**] “An Architecture for Intelligent Reactive Systems”, Leslie Pack Kaelbling, *SRI Tech Note 400*, October.

[Kaelbling 88] "Goals as Parallel Program Specifications", Leslie Pack Kaelbling, *AAAI-88*, St Paul, MN, August, 60–65.

[Kanayama and Miyake 86] "Trajectory Generation for Mobile Robots", Yutaka Kanayama and Norihisa Miyake, *Robotics Research 3*, Faugeras and Giralt eds, MIT Press, 333–340.

[Kriegman and Binford 88] "Generic Models for Robot Navigation", David J. Kriegman and Thomas O. Binford, *Proceedings IEEE Robotics and Automation*, Philadelphia, PA, April, 746–751.

[Lin, Mitchell, Philips and Simmons 89] "A Case Study in Robot Exploration", Long-Ji Lin, Tom M. Mitchell, Andrew Philips and Reid Simmons, *CMU-RI-TR-89-1*, January.

[Maes 89] "The Dynamics of Action Selection", Pattie Maes, *AAAI Spring Symposium on AI Limited Rationality*, Stanford, CA, March, 87–91.

[Mann, Hamel and Weisbin 88] "The Development of an Intelligent Nuclear Maintenance Robot", Reinhold C. Mann, William R. Hamel and Charles R. Weisbin, *Proceedings IEEE Robotics and Automation*, Philadelphia, PA, April, 621–623.

[Minsky 86] "Society of Mind", Marvin L. Minsky, *Simon and Schuster*, New York.

[Moravec 83] "The Stanford Cart and the CMU Rover", Hans P. Moravec, *Proceedings of the IEEE*, 71(7), July, 872–884.

[Moravec 84] "Locomotion, Vision and Intelligence", Hans P. Moravec, *Robotics Research 1*, Brady and Paul eds, MIT Press, 215–224.

[Nilsson 84] "Shakey the Robot", Nils J. Nilsson, *SRI Tech Note 323*, April.

[Noreils and Chatila 89] "Control of Mobile Robot Actions", Fabrice R. Noreils and Raja G. Chatila, *Proceedings IEEE Robotics and Automation*, Scottsdale, AZ, May, 701–707.

[Sarachik 89] "Characterising an Indoor Environment with a Mobile Robot and Uncalibrated Stereo", Karen B. Sarachik, *Proceedings IEEE Robotics and Automation*, Scottsdale, AZ, May, 984–989.

[Thorpe, Hebert, Kanade and Shafer 88] "Vision and Navigation for the Carnegie-Mellon Navlab", Charles Thorpe, Martial Hebert, Takeo Kanade and Steven A. Shafer, *IEEE Trans. PAMI*, 10(3), May, 362–373.

[Turk, Morgenthaler, Gremban and Marra 88] "VITS-A Vision System for Autonomous Land Vehicle Navigation", Matthew A. Turk, David G. Morgenthaler, Keith

D. Gremban and Martin Marra, *IEEE Trans. PAMI*, 10(3), May, 342–361.

[Viggh and Flynn 88] “Infrared People Sensors for Mobile Robots”, Herbert E. M. Viggh and Anita M. Flynn, *SPIE Vol. 1007 Mobile Robots III*, Cambridge, MA, November, 391–398.

[Walter 50] “An Imitation of Life”, W. Grey Walter, *Scientific American*, 182(5), May, 42–45.

[Walter 51] “A Machine that Learns”, W. Grey Walter, *Scientific American*, 185(2), August, 60–63.

[Wells 87] “Visual Estimation of 3-D Line Segments from Motion—A Mobile Robot Vision System”, William M. Wells III, *AAAI-87*, Seattle, WA, July, 772–776.

A Example Behavior

To demonstrate the behavior language, we present the complete code of the behavior which controls Seymour’s mobile base. It is set up to take commands in the form of a carrot dangling on a stick somewhere out in front of it. The base moves towards that carrot, but if it has not received any new reasons to move in the meantime, it completely halts after 3 seconds. If at any point a current overload is detected in either the rotation motor, or the translation motor, it probably means that the creature has hit something, and so an evasive maneuver can be made. Also, the base behavior is willing to live with another behavior which sometimes commands it to stop altogether.

In the code below, the symbols starting with “\$” indicate constants. There are three important inputs: `stop` to stop the robot, and `heading` and `move` to control where the carrot is hung. The other four inputs to the behavior, `delta-tv`, `delta-rv`, `set-tv` and `set-rv`, provide ways either directly, or incrementally, to control the translational or rotational velocities of the base. There is a single `status` output, which advertises to whomever may want to listen to it, whether the base is active or not.

The primitive operations called by this behavior are:

`update-base-status` which gets the most current information on motor currents from the base and caches them into the processor on which the behavior language runs. It also keeps a most recent maximum value of such currents.

`base-status` which returns the particular (specified by an argument) motor current, or recent maximum motor current requested.

`clear-base-status` which clears the most recent maximum current for a particular (specified by an argument) motor current.

`enqueue1` which sends a one-argument command down to the base.

`enqueue2` which sends a two-argument command down to the base.

There are nine toplevel processes within the base controller. Their order of specification is not important. They happen to be in the following order:

1. Update the motor current caches at 25Hz.
2. Whenever the translational motor exceeds a certain current, take evasive action by backing up. It also triggers the `halted` monstable for a second.
3. Whenever the rotational motor exceeds a certain current, take evasive action by turning in the opposite direction a little. It also triggers the `halted` monstable for a second.
4. Sanitize any external commands to set the translation velocity to obey maximum and minimum constraints.
5. Sanitize any external commands to set the rotation velocity to obey maximum and minimum constraints.
6. When there is an internally generated `stop`, the base controller may want to back up the robot, but during some small time it should not listen to outside requests to move the robot as they may have been generated before the effect of the robot stopping has trickled through the sensor chain. This process renames `external move` to the internal `imove` command unless there has recently been a `halt` asserted.
7. This is really the main process of the behavior. It mediates between stop commands, motion commands, and turn commands that are received while the robot is halted. `stop` commands have the highest priority. Unlike the `halt` commands that are generated by motor current overloads, this process gently decelerates the vehicle. The next priority are motion commands, which are the carrots dangling on sticks ahead of the robot. The lowest priority is responding to a turn in place command given when the robot is halted for some reason. By grouping these three subordinate `whenever` clauses inside an `exclusive` clause, whenever one is active it starves out the other two ensuring there will be no conflict between them.
8. Output the status (moving or halted) at 5Hz.
9. Stop all motion unless some external behavior has generated a request for motion within the last 3 seconds. By checking for an `imove` command having been received within an `exclusive` construct the `(delay 3.0)` gets reset if there is any such command. It generates a soft stop command that will be handled by process number 7.

```
(defbehavior base 0
:decls ((heading :init 0)
        (halted :monostable 1.0)
        (tvel :init $translate-default-velocity)
        (rvel :init $rotate-default-velocity)
        (base-state :init $halted))
:inputs (stop heading move delta-tv delta-rv set-tv set-rv)
:outputs (status)
```

```

:processes
((whenever t (update-base-status))

;; 2. translational motor current overload checking
(whenever (> (base-status :tcm) $translate-max-current)
  (enqueue1 :ki)
  (trigger halted)
  (setf tvel $translate-default-velocity)
  (setf rvel $rotate-default-velocity)
  (delay 0.2)
  (clear-base-status :tcm)
  (clear-base-status :rcm)
  (output heading 0)
  (if (> last-move 0)
    (output imove (cms -10))
    (output imove (cms 10)))))

;; 3. rotational motor current overload checking
(whenever (> (base-status :rcm) $rotate-max-current)
  (enqueue1 :ki)
  (trigger halted)
  (setf tvel $translate-default-velocity)
  (setf rvel $rotate-default-velocity)
  (delay 0.2)
  (clear-base-status :tcm)
  (clear-base-status :rcm)
  (if (> last-heading 0)
    (output heading (degrees -15))
    (output heading (degrees 15)))
  (output imove 0))

;; 4. translational velocity maintenance
(exclusive
  (whenever (received? set-tv)
    (setf tvel
      (max 0
        (min set-tv $translate-max-velocity)))))
(whenever (and (received? delta-tv) (delay 0.5))
  (setf tvel
    (max 0
      (min (+ tvel delta-tv)
        $translate-max-velocity)))))

;; 5. rotational velocity maintenance
(exclusive
  (whenever (received? set-rv)
    (setf rvel
      (max 0
        (min set-rv $rotate-max-velocity)))))
(whenever (and (received? delta-rv) (delay 0.5))

```

```

(setf rvel
  (max 0
    (min (+ rvel delta-rv)
      $rotate-max-velocity))))

;; 6. external generated moves are ignored after
;;   a halt but internal ones are used to get out
;;   of bad situations
(whenever (and (received? move) (not halted))
  (output imove move))

;; 7. command processing -- tied together so
;;   that a stop discards concurrent move
(exclusive
  (whenever (received? stop)
    (enqueue2 :ta $translate-halt-acceleration)
    (enqueue2 :ra $rotate-halt-acceleration)
    (enqueue1 :th)
    (enqueue1 :rh)
    (trigger halted)
    (setf base-state $halted)
    (setf tvel $translate-default-velocity)
    (setf rvel $rotate-default-velocity))
  ;; lower priority
  (whenever (received? imove)
    (enqueue2 :ta $translate-acceleration)
    (enqueue2 :ra $rotate-acceleration)
    (enqueue2 :tv tvel)
    (enqueue2 :rv rvel)
    (setf last-heading heading)
    (setf last-move imove)
    (setf base-state $moving)
    (if (> heading 0)
      (enqueue2 :r< heading)
      (if (< heading 0)
        (enqueue2 :r> (- heading)))))
    (setf heading 0)
    (if (> imove 0)
      (enqueue2 :t> imove)
      (if (< imove 0)
        (enqueue2 :t< (- imove)))))
  ;; lowest priority
  (whenever (and (received? heading) halted)
    (enqueue2 :ra $rotate-acceleration)
    (enqueue2 :rv rvel)
    (setf last-heading heading)
    (setf last-move 0)
    (setf base-state $moving)
    (if (> heading 0)
      (enqueue2 :r< heading)

```

```

        (if (< heading 0)
            (enqueue2 :r> (- heading))))
    (setf heading 0)))

;; 8. tell the world about the base status
(whenever (delay 0.2) (output status base-state))

;; 9. time out if there are no externally generated
;;    motions
(exclusive
  (whenever (received? imove)
    (nothing))
  ;; if haven't heard anything for a while
  ;; then cautionary stop
  (whenever (delay 3.0)
    (if (= base-state $moving)
      (output stop $halted)))))
))

```


Swarm Intelligence in Cellular Robotic Systems

Gerardo Beni and Jing Wang

College of Engineering
University of California, Riverside
Riverside, California 92521, USA

Abstract

Cellular Robotic Systems are capable of 'intelligent' behavior. The meaning of this intelligence is analyzed in the paper. We define robot intelligence and robot system intelligence in terms of unpredictability of improbable behavior. The concept of unpredictability is analyzed in relation to (1) statistical unpredictability, (2) inaccessibility, (3) undecidability, (4) intractability, and (5) non-representability. We argue that the latter two type of unpredictability, when exhibited by systems capable of producing order, can result in a non-trivial, different form of intelligent behavior (Swarm Intelligence). Engineering problems related to Swarm Intelligence are mentioned in relation to Cellular Robotic Systems which consist of collections of autonomous, non-synchronized, non-intelligent robots cooperating to achieve global tasks.

1. Introduction

1.1 The Concept of CRS

As a simplified model of general distributed robotic systems, the concept of Cellular Robotic Systems (CRS) has been introduced in [1]. A CRS consists of a large (finite) number of robots and operates on a finite n-dimensional cellular space under distributed control. No system wide centralized mechanism, synchronous clock or shared memory is assumed. Limited communication exists only between adjacent robots. On one hand, these robots have to operate autonomously; on the other hand, they have to cooperate to accomplish predefined global tasks. Detailed discussion on CRS has been presented in [2] and potential applications of CRS are shown in [3].

The advantages of cellular robotic systems are:

- (1) As compared with centralized robotic systems designed for the same task, each robotic unit of a CRS is of much lower complexity. The robotic units can be modularized, mass produced, interchangeable and (maybe) disposable.
- (2) Since CRSs are under distributed control, and since in general they are highly redundant, systems based on CRS are highly reliable. These systems can be designed to survive through various kinds of disturbances and to have the ability to dynamically adapt to their working environment. This is crucial in situations where very high reliability is required.
- (3) Since a CRS is essentially a massively parallel processing system, its collective computational power makes it attractive to carry out robotic tasks that are impossible for any single robot.

1.2 Closely Related Areas

One of the areas from which the CRS research arose is the theory of cellular automata [4]. The cellular model was introduced by Von Neumann in the context of machine reproduction [5]. Since then it has been used to model various physical and biological phenomena such as pattern growth [6], perception [7] and language recognition [8]. Hardware implementations range from the Connection Machine [9] to the MIT CAM board that can be used with an IBM PC [10].

As compared with systems based on the cellular automata model, which are essentially *homogeneous, static* and *synchronous*, a CRS employs intelligent machines (robots) capable of processing both *information* and *matter*. In the theory of cellular automata, no conceptual distinction has been made between the two entities. Matter, if any, has been embedded into the set of states of the system. Therefore, general models of cellular automata are (though capable) not the best to capture characteristics of the physical activities of actual robotic systems. Taking matter processing into account, the CRS model puts restrictions on cellular automata to reflect the material nature of the cell elements. Because of this essential difference, CRS based systems have to be *heterogeneous, kinetic* and *asynchronous*.

Another area that has great influence on CRS research is distributed computing[11]. It has been observed that there is a collection of common "sub" (low level) protocols upon which various "main" protocols operate. Moreover, these main-protocols are in general independent of, or at least may be isolated from, these supporting protocols. It is not surprising to realize that these supporting protocols are often directly related to many current research issues of distributed computing. A protocol that governs coordinated robot motion without collision exemplifies protocols in this category (This problem will be discussed in Section 4). Many classical problems in distributed computing, e.g., *mutual exclusion, leader finding, agreement* and *majority voting* can all find their "applications" in the CRS theory.

1.3 Paper Organization

The next section defines Swarm Intelligence. The definition requires a discussion of the concept of 'unpredictability' which is the main contribution of the paper. Implications and engineering issues in CRS are mentioned in the last two sections. The paper includes conjectures and tentative results and is not intended to be mathematically rigorous.

2. "Swarm" Intelligence

2.1 Robot Intelligence

CRS are capable of 'intelligent' behavior in the following sense.

First we specify the meaning of some basic terms as used in this paper. "Machine" is defined as an entity capable of mechanical behavior (i.e. transfer and/or processing matter/energy). "Automaton" is defined as an entity capable of informational behavior (i.e. transfer and/or processing information). "Robot" is defined as an entity capable of *both* mechanical and informational behavior (i.e. a robot is both an automaton and a machine). These definitions are somewhat different from the common usage, but they help avoiding confusion since they contain only the two well-defined concepts of "matter/energy" and "information". An additional specification is also necessary for the term 'environment'. In what follows environment is intended to include everything which is not the system itself.

Since information represents states of matter, it is clear that we can have 'pure' automata, i.e. entities capable only of transferring and/or processing information. On the other hand, strictly speaking, it is impossible to realize a 'pure' machine, i.e. an entity that transfer and/or processes matter only, i.e. without transferring and/or processing also information, since mechanical states are states and thus information. However, it is convenient to separate, conceptually the two functions as we have done in our definitions. We may in fact assume, e.g. that the information change is negligible in comparison with the mechanical change produced by the machine.

Next we define an "Intelligent robot" as:

Def 2.1.1

A robot whose behavior in response to a change in the environment external to the robot, is neither random nor predictable (in the sense specified below) from physical measurements of the environment.

To satisfy this definition, we shall see later that an intelligent robot must contain an internal model, or representation, which is, generally, not accessible to the external observer. This internal model, plus the external environment, determines the robot's behavior via an internal algorithm. (Modifications of the model and even the algorithm are instances of internal robot behavior.)

It follows that memory is a necessary attribute of an intelligent robot. But it is not sufficient-- common hysteretic systems are predictable from external measurements -- *since the robot must be capable of modifying its internal state via its internal state, plus the external environment, if it has to be unpredictable from the external environment, as required by our definition.* The term 'unpredictable' has been left ambiguous in the definition of intelligent robot. Its meaning is crucial to the definition but to specify it we need to introduce systems of robots, which is done in the next section.

2.2 Robot System Intelligence

The previous definitions refer to a generic entity, regardless of its plurality, thus they apply to systems as well. Systems, however, have interesting properties in relation with robotic behavior. First, no system containing only 'automata' can be a robot, but some systems containing only 'machines' may be robots. These conclusions follow from our definitions of automaton and machine (Sec. 2.1) which in turn are based on the concepts of matter/energy and information.

In fact, a collection of 'pure' machines can transfer and/or process information. This happens whenever the 'machines' encode information as patterns of themselves. (Which is basically how biological (e.g. genetic) information is transferred and/or processed.) Thus, a collection of machines can be a robot.

This fact opens up the possibility of building robots out of simpler units, i.e. machines, which individually are not capable (or have limited capability) of processing information but collectively can transfer and/or process both matter and information. Several questions along these lines have been raised and answered in [12,13]. Here we wish to go one step further and focus on the meaning of building *intelligent* robots out of non-intelligent robots.

Intelligent robot was defined in Sec. 2.1 in terms of unpredictability of behavior. The actual behavior and unpredictability were not specified. First, we specify behavior in terms of the relation of the intelligent robot to (ordered) patterns of matter (i.e. arrangements of material objects, in contrast with patterns of states of matter, i.e. pure information) in the external environment. In terms of such patterns there can be basically two types of intelligent behavior: pattern analysis (e.g. recognition) and pattern synthesis (e.g. pattern generation). The first type can be accomplished by 'pure' automata systems, while the latter can only be accomplished by intelligent robots. Thus, we restrict our attention to this type of intelligent behavior, i.e. we will be concerned with the intelligence of:

Systems of non-intelligent robots exhibiting collectively intelligent behavior evident in the ability to produce unpredictably specific (=not in a statistical sense) ordered patterns of matter in the external environment. ()*

We note that since such systems synthesize patterns (i.e. produce order) they must be dissipative. In the simplest case this condition is satisfied if the system is energetically open, e.g. the units (non-intelligent robots) of the system could contain energy sources. Without loss of generality we may assume that this is the case.

2.3 Unpredictability

In (*) we use the term 'specific' (=not in a statistical sense). Without this qualification, (*) would appear to be satisfied also by synergetic [14] self-organizing systems. It is well known that macroscopically ordered states can be formed by non linear dynamic systems depending on the growth rate of their collective modes. In some cases, even without quantum fluctuations, the future state-space path of the system cannot be predicted. This unpredictability is due to the fact that such systems are extremely sensitive, in their further development, to initial conditions, which may result in ordered states. The self-organizing behavior of these systems, however, is essentially statistic. Non equilibrium, non-linear statistical physics is a powerful tool for studying the statistical formation of ordered patterns in self-organizing systems. Our interest here is *not* in systems composed of *very many systems* whose properties are calculated *statistically*. In CRS for example, we are interested in a sort of 'not-so-many-body problem', of the order of ~100 robot, which we would like to describe non-statistically.

Thus, although unpredictability can occur without internal models (i.e. even if the system is totally accessible), this type of purely stochastic unpredictability is insufficient to produce intelligent behavior as defined in sect. 2.1.

In that section we stated that unpredictability from the external environment requires the ability to modify one's internal state via its internal state (and the external environment). This ability is not possessed by a large number of systems which otherwise appear to satisfy (*). For example, systems of biological cells such as plants seem to satisfy the definition. However, for these systems the 'unpredictability' is due simply to the *physical* inaccessibility of the internal model (the genetic code). This type of unpredictability is not intrinsic since we might envision methods of determining the genetic structure from physical measurements. Obviously, these measurements would be extremely complex and probably would require a long time but, since the genetic code cannot alter itself (we exclude mutations since they usually occur on a long time scale) it can be eventually accessed, making the system actually predictable. Similar considerations apply to any system whose unpredictability is due simply to physical inaccessibility of its internal model.

More interesting are systems for which the internal state is physically accessible but it remains unpredictable. Such systems do exist; in fact, if we turn to systems composed of pure automata units, we can immediately find this type of unpredictability. Indeed, it is well known that the infinite time behavior of universal automata is unpredictable [15]. Thus, for example, a 'mechanical' universal automaton could be regarded as an unpredictable system composed of non-intelligent robot units. And since it is capable of producing ordered patterns, it satisfies (*) .

Although a mechanical universal automaton (e.g. a mechanical 'game of life' cellular automaton) appear to provide a sort of 'proof of existence' for systems satisfying (*), the type of unpredictability of universal automata is very limited, in the following sense.

First of all, what is precisely meant by unpredictability of universal automata ? The argument runs as follows [16]. The only way to predict the behavior of any system is by computation on a universal automaton. This computation, in general, reduces to a direct simulation if the system whose behavior is to be predicted is itself capable of universal computation. But in this case the speed of this simulation differs only by about a constant factor from the speed at which the simulated system itself evolves. Thus, the infinite time behavior of a system capable of universal computation is in general unknowable in any finite time, and thus the problem is formally undecidable.

Thus, what is meant by unpredictability in [16] is essentially undecidability. But, what of systems whose behavior is unpredictable not just 'at infinity' but 'at every step' ? That is systems that cannot be predicted, at any step (=during the time that it takes to create a new pattern), by any computer ?

Clearly these step-wise unpredictable systems are the ones of interest for intelligent behavior, since we are interested in intelligent behavior over a range of time and not just at infinity.

Step-wise unpredictability of a system can of course exist whenever (some of) the components themselves are step-wise unpredictable. This is a trivial case. According to (*) we are interested in systems whose components are predictable but globally they are unpredictable.

This can be realized in two ways. In the first, the global system is, strictly speaking predictable, but not in real time. (Here 'in real time' means ' during the time that the system takes to produce a new pattern.) In other words, the problem of predicting is 'intractable'.

In the second case, the global system is non-predictable (regardless of speed of computation) for it is essentially not-representable externally. Both cases are of conceptual and practical interest. Both types of unpredictability are included in the definition of Swarm Intelligence (Def. 2.2.1) below. The second case will be discussed in detail elsewhere. Here we make some remarks on the 'intractable case'.

The operations of the traditional 'intelligent' robot are :

- (1) sensing the environment,
- (2) computation ,
- (3) actuation.

The three operations are carried out sequentially. Assume that such a robot, A, in step (3), produces patterns. These patterns, along with the environment are observed by a second robot, B, whose step (2) is carried out by universal computer which knows the initial state

of A's computer. Then A is predictable by B in real time: before A can produce a new pattern, B will have calculated it.

In contrast, consider the operation of a collection C of non-intelligent robots synthesizing patterns of themselves (e.g. mobile robots arranging themselves in patterns as in CRS). For the robotic system C, steps (2) and (3) happen in parallel. If the observer robot B knows the initial state of C can it predict, in real time, the pattern produced by C? It depends on the speed of each robot unit in C, since B must simulate the dynamics of C. Assume that for C, step (3) is much slower than (2), which is plausible, and let T_c be the upper limit on step (3) for any robot unit. (since they operate asynchronously, without shared clock, their operation times will vary, but an upper limit can be assumed). Thus, T_c is the 'real time' in which C creates a new pattern. The universal computer B must predict this pattern, from the initial state of C and the external environment, in a time T_b : $T_b < T_c$.

Assume first that B's computer has a Von Neumann architecture. Since T_c does not depend on the number N of robot units in C, and thus does not depend on the number of states N_c of C, whereas T_b depends on N_c , we can always obtain $T_c < T_b$ if N is sufficiently large. Thus, C is unpredictable. Assume now that B's computer has any architecture, e.g. a cellular automaton architecture, and it is set up to simulate C's evolution, from its initial state. But since the evolution of C includes its dynamics (which depends, among other things, on the internal clocks of the robots in C, which may run deterministically but at unpredictable speeds) it cannot be identically simulated. Any approximation to the dynamics of the clocks will result in computations of complexity of the order of N_c . Thus if N_c is large enough, the time T_b of these computations will also be $T_b' > T_c$.

In any case, in practical applications, unpredictability is reached very quickly for relatively small N. For example, if $T_c = 1\text{msec}$ and $N = 100$, $N_c \sim 2^{100}$ which is a problem beyond current supercomputer computational power.

Thus a class of robot systems, both conceptual and practical, can be defined exhibiting an interesting type of intelligent behavior :

Def 2.2.1 ("Swarm Intelligence")

Systems of non-intelligent robots exhibiting collectively intelligent behavior evident in the ability to unpredictably (#) produce specific (=not in a statistical sense) ordered patterns of matter in the external environment.

Unpredictable is meant as globally 'intractable' or 'externally not-representable'.

3. Comparisons

Since 'unpredictability' is the key concept in Def 2.2.1 it is useful to compare the different types of unpredictable systems mentioned so far. Figure 1 shows schematically the relation between unpredictable systems. The pure statistical unpredictability (of e.g. stochastic systems) can produce (under non-equilibrium conditions) ordered patterns, but only statistically. Other forms of unpredictability depends in some way on the inaccessibility of internal states. Intrinsic (e.g. physical) inaccessibility (such as the genetic code of biosystems) has limited interest. Inaccessibility due to lack of computational capability lead us to distinguish between undecidability and step-wise unpredictability. The latter can lead, in distributed dynamical systems, to intractability or external not-representability--these situations constitutes the unpredictability of "swarm intelligence".

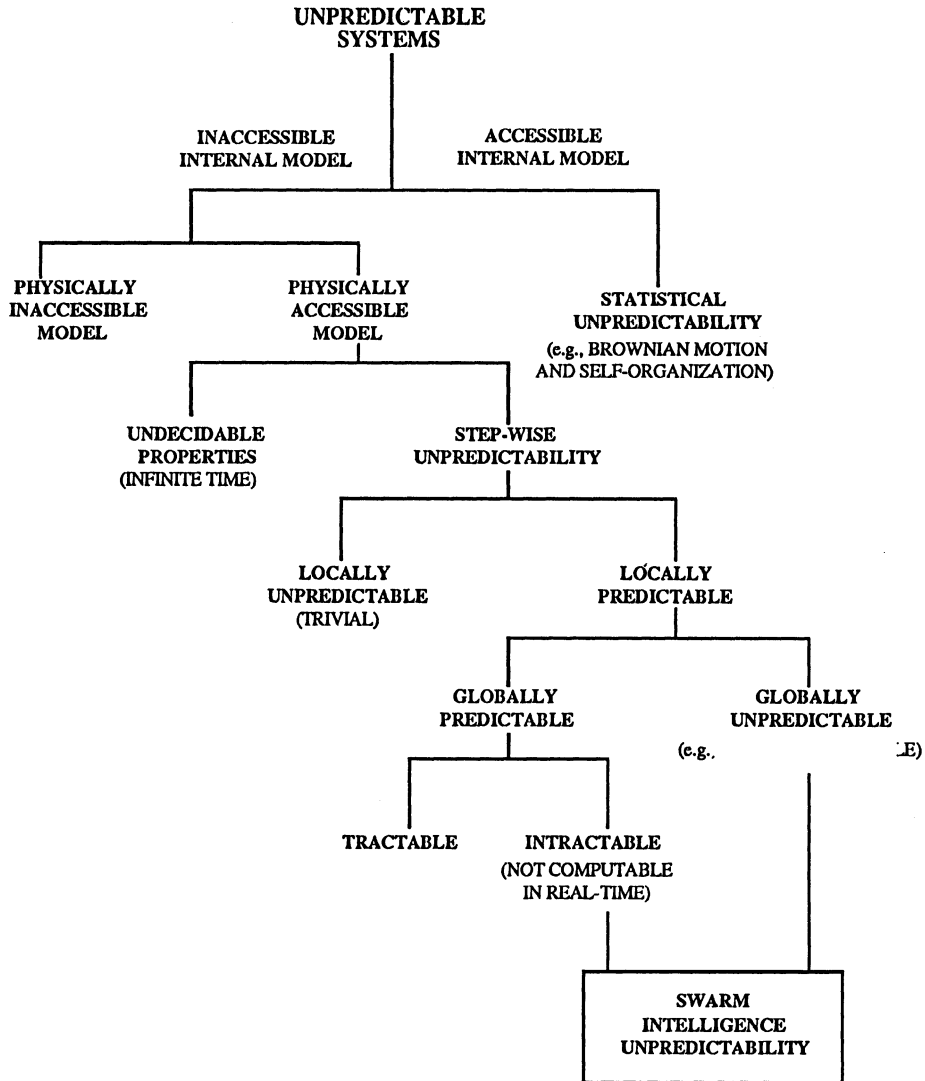


Figure 1. Relationship between unpredictabilities

We can also look at Swarm Intelligence in relation to its cause-effect characteristics. In Figure 2 we position Swarm Intelligence with respect to four other groups of systems characterized by having either predictable or unpredictable means of producing either 'probable' or 'improbable' effects, e.g. random or ordered patterns, respectively. Examples of systems that (1) predictably produce 'probable' results are ordinary dynamic systems and also non-linear systems deterministically producing chaotic behavior. Systems that (2) predictably produce 'improbable' results are machines, deterministic automata and non-intelligent robots. Typical systems that (3) 'unpredictably' produce the 'probable' are stochastic systems producing random output, e.g. Brownian motion. Finally, (4) systems

that unpredictably produce the 'improbable' range from statistically self-organizing systems, to cellular automata, to Swarm Intelligent systems, depending on the type on unpredictability, as discussed in Section 2.3. In the context of Figure 2, Swarm Intelligence may be concisely defined as 'the intractable or non-representable creation of the improbable'.

		EFFECT	
		•PROBABLE	•IMPROBABLE
CAUSE	• PREDICTABLE	-ORDINARY DETERMINISTIC DYNAMIC SYSTEMS (INCL. CHAOS)	-MACHINE -AUTOMATON -NON-INTELLIGENT ROBOT
	• UNPREDICTABLE (RANDOM) (PHYSICALLY INACCESSIBLE) (UNDECIDABLE) (INTRACTABLE) (EXTERNALLY NON- REPRESENTABLE)	-BROWNIAN MOTION	-STATISTICAL SELF-ORGANIZATION -GENETIC CODE -SOME CELLULAR AUTOMATA SWARM INTELLIGENCE

Figure 2. Unpredictability in relation to its cause-effect characteristics

4. Conclusion

Swarm Intelligence appears as the interplay of computation and dynamics. Theoretically, the two areas have only in the last few years begun to merge, notably in the line of research initiated by S. Wolfram. Most of the emphasis has so far been placed on cellular automata. Studies of their undecidability and relation to pattern formation are numerous.

However, there are no extensions to robot systems yet (other asynchronous systems have also been generally neglected). Since robots are the systems which, by definition, merge computation and dynamics, we expect that a theoretical foundation of Robotics Science will eventually emerge around a theory of dynamical/computational systems.

Meanwhile, engineering problems of systems exhibiting Swarm Intelligence, such as CRS, may help shed some light on some of the fundamental issues. Examples of such problems in CRS are:

- (1) extension of distributed control protocols to 'knot' and 'boundary' problems[2];
- (2) subprotocols [17,18];
- (3) active and passive 'swarm' architectures[19];
- (4) specific problems such as self-assembly [20] and manufacturing [19];
- (5) control and repair via environment sensing;

Swarm intelligent' as defined in 2.2.1 is likely to be exhibited by a large number of animal societies, e.g. ant colonies. Thus, we might expect that the engineering of Swarm Intelligence, which appears at the moment, only at the beginning of a rather arduous task, will be likely to benefit from a 'bionic' perspective.

5. Acknowledgements

We thank Susan Hackwood for critical discussions on the manuscript, and Beatrix Zimmermann for her contributions to the CRS research. We are also thankful to Alex Meystel for insights on the concept of Swarm Intelligence and Harry Stephanou for discussions on CRS. This work has been partially funded by NSF under contract No.0842415.

6. References

1. G.Beni, "Concept of Cellular Robotic Systems", Presented at the 3rd IEEE Symposium on Intelligent Control, Arlington, Virginia, August 24-26, 1988.
2. J.Wang and G.Beni, "Pattern Generation in Cellular Robotic Systems", Presented at the 3rd IEEE Symposium on Intelligent Control, Arlington, Virginia, August 24-26, 1988.
3. S.Hackwood and J.Wang, "Application and Engineering of Cellular Robotic Systems", Presented at the 3rd IEEE Symposium on Intelligent Control, Arlington, Virginia, August 24-26, 1988.
4. E.F.Codd, "Cellular Automata", Academic Press, New York, 1968.
5. J.Von Neumann, "Theory of Self-Reproducing Automata", Edited and completed by Burks, A.W., University of Illinois Press, Urbana and London, 1966.
6. S.Wolfram, "Universality and Complexity in Cellular Automata", *Physica* 10D, 1984, pp.1-35.
7. A.Rosenfeld, "Picture Languages: Formal Model for Picture Recognition", Academic Press, New York, 1979.
8. A.R.Smith, "Real-Time Language Recognition by One-Dimensional Cellular Automata", *Journal of Computer and System Sciences*, 6, pp. 233-253, 1972.
9. W.D.Hills, "The Connection Machine", MIT Press, Cambridge, MA, 1987.
10. T.Toffoli and N.Margolus, "Cellular Automata Machines: a New Environment for Modeling", MIT Press, Cambridge, MA, 1987.
11. Cornafion, "Distributed Computing Systems", Elsevier, Amsterdam, 1985.
12. J.Wang and G.Beni, "SCRS: CRS with Stationary Robots and Passive Objects", in preparation.
13. J.Wang and G.Beni, "Self-Organizing Robots and Kinetic Pattern Generation", CRSM Internal Report, June 1987.
14. e.g. H. Haken, "Advanced Synergetics", Springer-Verlag (Berlin, 1983).
15. e.g. J. Hopcroft and J. Ullman, "Introduction to Automata Theory, Languages and Computation" (Addison-Wesley, Reading, Mass. 1979).
16. S. Wolfram, "Some Recent Results and Questions about Cellular Automata", in "Dynamical Systems and Cellular Automata", Ed. by J. Demongeot, E. Goles, and M. Tchuente, (Academic Press, 1985).
17. J.Wang, "CRS: Problems Related to Distributed Computing", in preparation.
18. B. Zimmermann, J. Wang, and G. Beni, "Distributed Computing Algorithms in Cellular Robotic Systems I: The Commit Problem", to be presented at the 12th IASTED Intl. Symp. on Robotics and Manufacturing, Santa Barbara, Ca., Nov. 12-15, 1989.

19. J. Wang and G. Beni, "Cellular Robot System with Stationary Robots and its application to Manufacturing Lattices" to be presented at 4th IEEE International Symposium on Intelligent Control, Albany NY, Sept. 25-27, 1989.
20. B. Zimmermann, "The pairing Problem in a Cellular Robotic System", Master's Thesis, Department of Computer Science, Univ. of Calif. Santa Barbara, 1988.

A Control Architecture for Cooperative Intelligent Robots

James S. Albus*
Robot Systems Division
National Institute of Standards and Technology
Gaithersburg, MD USA

Abstract

Intelligent behavior requires a control system architecture that ties together actuators, sensors, sensory processing, task decomposition, world modeling, goal selection, and value judgments into an integrated system. A hierarchically layered architecture with horizontal communications within layers is proposed.

Intelligent cooperative behavior requires common goals, common strategies, agree upon division of labor, and a common view of the world. To the extent that all individuals in a population are working on the same top level input command, they share a common goal. To the degree that all have similar task vocabularies and control programs resident in their task decomposition modules, there exists a basis for common strategy. To the extent that all agree on the structure of the command tree, there exists a basis for division of labor. To the degree that individuals have the same information resident in their world model, they share a common view of the world. When all these exist simultaneously, there exists the basis for intelligent cooperative behavior. Communication is a mechanism for maintaining a common goal, strategy, division of labor, and world view.

1. Introduction

During the past decade, the problem of cooperative intelligent agents has become a substantial subfield of artificial intelligence. Distributed artificial intelligence, sometimes known as distributed problem solving [1] is concerned with distributed planning and control as well as with distributed data and multi-processor computing architectures. The techniques of distributed problem solving have been most extensively studied in the areas of speech understanding [2], manufacturing process scheduling [3,4] air traffic control [5,6], and multiple autonomous undersea vehicles [7].

*Contribution of the United States Federal Government, not subject to copyright.

Nilsson [8] has suggested that distributed problem solving may be crucial to understanding of artificial intelligence. There are many reasons to believe that it is crucial to understanding of biological intelligence as well. The brain is not a single computer nor is mind a single process [9]. The brain is a community of billions of neurons, and the mind a collection of millions of processes all interconnected and independently executing simultaneously. The brain and nervous system is one vast distributed processing system with interconnected modules for task decomposition, world modeling, sensory processing, and value estimating.

Furthermore, problem solving in the brain is not limited to the individual. The nature of the world is such that few creatures can survive alone, and there are many survival advantages in cooperation between individuals. There is safety in numbers, and many tasks, from hunting to escaping predation, are more likely to succeed if pursued by cooperating groups rather than individuals.

Virtually all animals exhibit some form of cooperative behavior. Fish swim in schools, birds congregate in flocks, grazing animals form herds, and many predators hunt in packs. The advantages are that the school or flock always has many eyes looking in many directions at once for danger. The herd can form a circle to ward off predators. The hunting pack has many members searching for prey and cooperating in its capture.

In higher animals, cooperative behavior is extensive, even pervasive. Parents cooperate in feeding offspring, protecting them from danger, and training them in the ways of the world. Among the most intelligent of creatures such as elephants, apes, and humans, families live together for years, and group interactions involve elaborate protocols including the establishment of rank in social dominance hierarchies.

In human societies, more efficiently organized social structures with extensive division of labor and well trained workers tend to be more efficient than less organized cultures, surpassing them economically and dominating them militarily and politically.

It is anticipated that artificially intelligent autonomous military vehicles will be most effective when deployed in groups.

2. Elements of Cooperation

Cooperation between intelligent individuals requires a set of commonalties. There must be a common goal, a common strategy or plan for achieving that goal, a common view or model of the world, an agreed upon division of labor, and a means for communication between the cooperating individuals.

Common goals

Among creatures of nature, all goals relate ultimately to survival and gene propagation. Survival requires detecting and escaping predators, and hunting for food. Propagation requires acquiring and defending territory, attracting a mate, defeating rivals, and feeding and defending offspring.

Where there exists common goals, such as to watch for and ward off predators, to track and kill prey, or to defend offspring, cooperation will be the predominant behavior. Where the goal of one individual conflicts with that of another, such as in the acquisition of territory, or attracting breeding partners, conflict and competition will predominate.

Common strategy

Without a common strategies, common goals cannot be achieved. A group cannot stay together unless all members move in roughly the same direction. A nesting pair will not be successful in raising a family unless they have a common strategy where one protects the nest while the other searches for food. A hunting pack will fail unless the members maintain contact and cooperate in the kill.

Division of labor

Cooperation implies that each member plays a role in a joint activity. Group tasks are decomposed into subtasks to be carried out by individual group members. In caring for young, one parent may hunt for food, while the other protects the nest. In defending young, one parent may attract attention to itself by attacking the intruder or feigning injury, while the other member sits motionless on the nest to provide camouflage.

In complex human enterprises such as manufacturing, construction, or military operations, division of labor is carried to a high degree with many hierarchical levels of decomposition and many different specialized activities.

Common world model

All intelligent creatures carry some internal representation of the external world, either explicitly or implicitly. Without a common world model representation of critical information such as the distribution of danger, the direction of escape, and the relative positions of friendly agents, cooperation cannot take place.

Communication

Communication is one means by which common goals, strategies, division of labor, and representation of the world can be established. An individual may use communication to announce its own identity, position, and state of mind, including intent. Communication can be used to sound an alarm at the presence of danger, or to announce the detection of prey. Communication can also be used to command or request action from others, or simply to inform others of events or state changes in the world.

Communication implies that information is: 1) encoded, 2) transmitted, 3) received, and 4) understood. Without all four, communication has not taken place. However, communication does not require intent on the part of the sender to send a message to the receiver. In fact, communication is often quite unintentional, and preventing an enemy from intercepting communication between friendly agents can be a major concern.

Language is the means by which information is encoded. Language has three basic components: vocabulary, syntax, and semantics. Vocabulary is the set of words in the language. Words may be represented by symbols. Syntax, or grammar, is the set of rules for stringing together symbols to form sentences. Semantics is the set of rules for combining words into meaningful patterns, or messages. Messages are sentences that convey useful information.

Communication and language are by no means unique to human beings. Virtually all creatures, even insects, communicate in some way, and hence have some form of language. For example, many insects transmit messages announcing their identity and position. Sometimes this is done acoustically, sometimes by smell. The goal may be to attract a mate, or to permit recognition and/or location by other members of a group. Of course, insects have very little other information to communicate, and hence their language has only one or two of what might be called words, with little or no grammar, and very simple semantics.

The encoding and transmission of information can take many forms, including body posture, gestures, facial expressions, hair erection, and acoustic signals generated by variety of mechanisms from stamping the feet or clapping the hands, to snorts, squeals, chirps, cries, and shouts. Depending on its complexity, a language may be capable of communicating many messages, or only a few. More intelligent creatures have a larger vocabulary and more complex grammar. They are also better able to understand and act on the meaning of messages.

Communication of danger signals improves the survival probability of all individuals in the group. However, communication is only advantageous to those individuals who are able to recognize the danger messages. Those benefit most who are the quickest and most discriminating in the recognition of danger messages, and most effective in responding with appropriate action.

In general, the benefit, or value, of communication is roughly proportional to the product of the amount of information contained in the message, multiplied by the ability of the receiver to understand and act on that information, multiplied by the importance of the act to survival and gene propagation. Greater intelligence enhances both the individual's and the group's ability to analyze the environment, to encode and transmit information about it, to detect messages, to recognize their significance, and act effectively on information received. Greater intelligence produces more complex languages capable of expressing more information, i.e. more messages with more shades of meaning.

In social species, communication also provides the basis for societal organization and division of labor. Communication of threats that warn of aggression can help to establish the social dominance hierarchy, and reduce the incidence of physical harm from fights over food, territory, and sexual partners. Communication of alarm signals indicate the presence of danger, and in some cases, identify its type and location. Communication of pleas for help enables group members to come to each other's assistance or defense. Communication between members of a hunting pack enable them to hunt more effectively by cooperating as a team in the tracking and killing of prey.

Among humans, the most basic form of communication is facial expressions, gestures, body language, cries, tone of voice, and pantomime. However, the human brain is capable of generating ideas of much greater complexity and subtlety than can be expressed through a few cries and gestures. In order to transmit messages commensurate with the complexity of human thought, human languages have evolved grammatical rules capable of stringing words from vocabularies consisting of thousands of entries into sentences which express ideas and concepts with exquisitely subtle nuances of meaning. During this evolutionary process, the human vocal apparatus has developed complex mechanisms for making a variety of sounds that can be strung together to generate an infinite number of messages in a spoken language.

In any species, language evolves to support the level of communication generated by the intelligence of that species. Cooperation and language are thus products of intelligence, and if we wish to fully understand either, we must first understand the mechanisms of intelligence itself.

Of course, our understanding of intelligence is far from complete and will remain so for a long time, but much is known that is relevant to the issue of cooperative behavior. For example, the fundamental elements of intelligence, and the basic functional relationships between them, are shown in Figure 1.

AN INTELLIGENT MACHINE SYSTEM

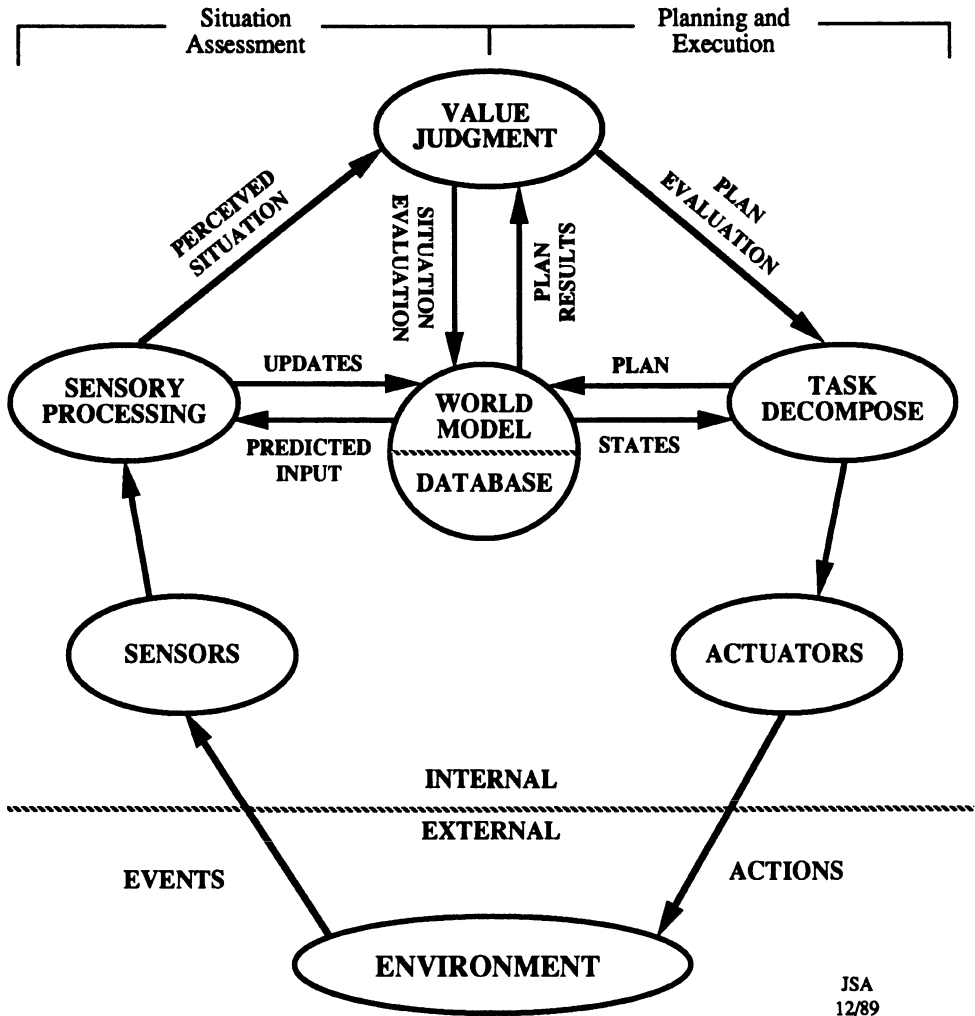


Figure 1. The elements of intelligence and the functional relationships between them.

3. The Elements of Intelligence

a. **Actuators** -- Within any intelligent system there are actuators which move, exert forces, and position arms, legs, hands, and eyes. Actuators generate forces to point sensors, excite transducers, move manipulators, handle tools, and steer locomotion. An intelligent system may have tens, hundreds, or even thousands of actuators, all of which must be coordinated in order to perform tasks and accomplish goals. Natural actuators are muscles and glands. Machine actuators are motors, pistons, valves, solenoids, and transducers.

b. **Sensors** -- Sensors may include visual brightness and color, tactile, force, torque, position, velocity, vibration, acoustic, range, smell, taste, pressure, and temperature measuring devices. Sensors may be used to monitor both the state of the external world and the internal state of the intelligent system itself. Sensors provide input to a sensory processing system.

c. **Sensory Processing** -- Perception takes place in a sensory processing system that compares observations with expectations generated by an internal world model. Sensory processing algorithms integrate similarities and differences between observations and expectations over time and space so as to detect events and recognize features, objects, and relationships in the world. Sensory input data from a wide variety of sensors over extended periods of time are fused into a consistent unified perception of the state of the world. Sensory processing algorithms compute distance, shape, orientation, surface characteristics, and material properties of objects and regions of space. Sensory processing may include recognition of speech and interpretation of language and music.

d. **Task Decomposition** -- Behavior is generated in a task decomposition system that plans and executes tasks by decomposing them into subtasks, and sequencing these subtasks so as to achieve goals. Goals are selected by a looping interaction between task decomposition, world modeling, and value judgment functions. The task decomposition system hypothesizes plans, the world model predicts the results of those plans, and the value judgment system evaluates those results. The task decomposition system then selects the plans with the best evaluations for execution. Task decomposition monitors the execution of task plans, and modifies existing plans whenever the situation requires.

e. **World Model** -- The world model is the intelligent system's best estimate of the state of the world. The world model includes a database of knowledge about the world, plus a database management system that stores and retrieves information. The world model also contains a simulation capability which generates expectations and predictions. The world model thus can provide answers to requests for information about the present, past, and probable future states of the world. The world model provides this information service to the task decomposition system,

so that it can make intelligent plans and behavioral choices, and to the sensory processing system, in order for it to perform correlation, model matching, and model based recognition of states, objects, and events. The world model is kept up-to-date by the sensory processing system.

f. Values -- The value system makes value judgments as to what is good and bad, rewarding and punishing, important and trivial. The value system evaluates both the observed state of the world and the predicted results of hypothesized plans. It computes costs, risks, and benefits both of observed situations and of planned activities. The value system thus provides the basis for choosing one action as opposed to another, or for pursuing one object and fleeing from another. Without a value system, any biological creature would soon be eaten by others, or destroyed by its own inappropriate actions. The value system also computes the probability of correctness and assigns believability and uncertainty parameters to state variables.

4. An Architecture for Intelligent Machine Systems

Each of the above elements of intelligent systems are reasonably well understood. However, intelligence is more than a set of disconnected elements. Intelligence requires an interconnecting system architecture that enables the various system components to interact and communicate with each other in an intimate and sophisticated way. A number of system architectures have been proposed, and a few have been implemented. [10, 11, 12, 13].

The particular architecture that will be discussed here has been partially implemented in a number of versions of the Real-time Control System (RCS) developed over the past 13 years at the National Institute for Standards and Technology (NIST formerly NBS). The first version of RCS was developed by Barbera for laboratory robotics and adapted by for the NIST Automated Manufacturing Research Facility (AMRF) [13]. RCS-2 was implemented and installed in the Horizontal machining workstation of the AMRF [14-18], and in the Cleaning and Deburring workstation [19]. RCS was adapted to the AMRF Vertical machining workstation [20, 21] and to the Inspection workstation [22]. The AMRF communications and database systems were developed [23], as well as the RCS emulator [24]. RCS-2 was adapted to the Army Field Material-handling Robot (FMR) [25].

RCS-3 has been, and is being, used on a number of additional projects, including the NBS/DARPA Multiple Autonomous Undersea Vehicle (MAUV) project and the Army TEAM (Technology Enhancement for Autonomous Vehicles) semi-autonomous land vehicle project. RCS-3 also forms the basis of the NASA/NBS Standard Reference Model Telerobot Control System Architecture (NASREM) [26] being used on the space station Flight Telerobotic Servicer.

The version of RCS described in this paper organizes the elements of intelligence defined in Section 3 into a six layer hierarchy of computing modules, partitioned into four vertically integrated sections: task decomposition (TD), world modeling (WM), sensory processing (SP), and value judgment (VJ). All the modules in this hierarchy are richly interconnected to each other by a fifth vertical section, the global memory (GM) which provides intermodule communication, as illustrated in Figure 2. In a biological brain, the function of the GM modules are provided by neurons as shown in Figure 3. In artificial systems, the physical implementation of global memory may be a common memory, a blackboard, or a message passing system, or some combination thereof.

Hierarchical versus Heterarchical

Figure 4 elaborates the architecture of Figure 2 so as to show both the hierarchical and heterarchical (horizontal) relationships involved.

The architecture is organized as a hierarchical tree in the sense that commands and status feedback flow hierarchically up and down the chain of command. It is also hierarchical in that sensory information is processed into increasingly higher levels of abstraction, and that information stored at various levels of resolution in the world model is organized hierarchically.

The levels in this hierarchical tree structure are defined by the decomposition of tasks into levels of spatial and temporal resolution. Spatial resolution is manifested in the span of control and in the resolution and range of maps. Temporal resolution is manifested in terms of loop bandwidth, sampling or state-change intervals. Temporal span is measured in length of historical traces, and future planning horizons.

High level commands, or goals, are decomposed both spatially and temporally through a hierarchy of control levels into strings and patterns of subcommands. Each task decomposition TD module represents a node in a command tree. Each command node receives input commands from one and only one supervisor, and outputs a temporal string of subcommands to one or more subordinate modules at the next level down in the tree. Outputs from the bottom level consist of drive signals to motors and actuators. Figure 5 shows a single branch of the hierarchical tree structure of Figure 4, seen from a high level looking down to a single motor unit.

The proposed architecture is, however, also heterarchical (or horizontal) in the sense that data is shared horizontally between heterogeneous modules at the same level. At each hierarchical level, the architecture is horizontally interconnected by wide bandwidth communication pathways between task decomposition TD, world modeling WM, sensory processing SP, and value judgment VJ modules.

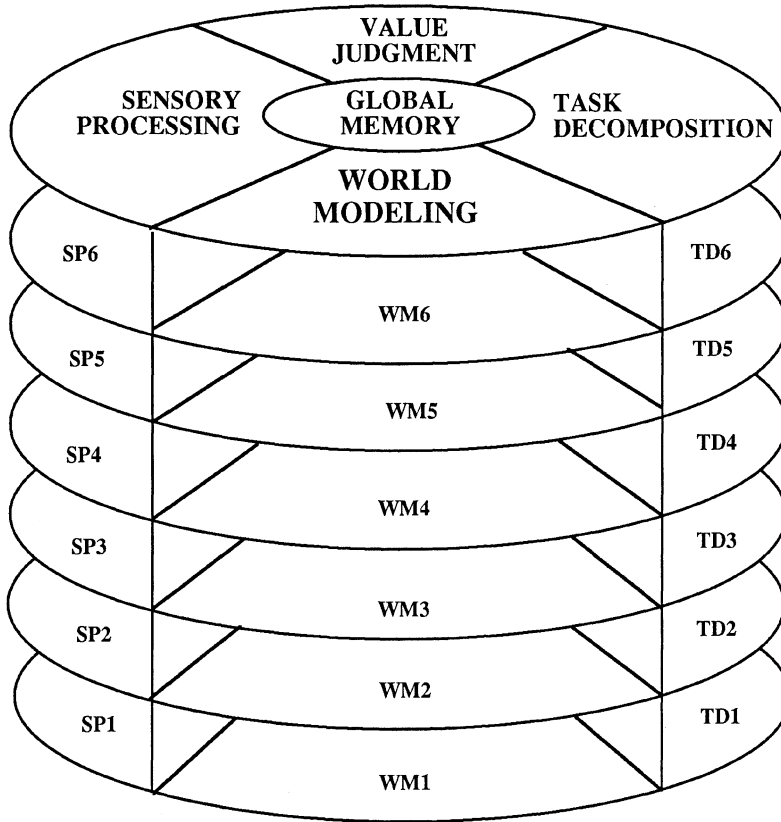


Figure 2. A six layer version of RCS partitioned into four vertically integrated sections: task decomposition TD, world modeling WM, sensory processing SP, and value judgment VJ. These are interconnected through the global memory GM which acts as a communications system.

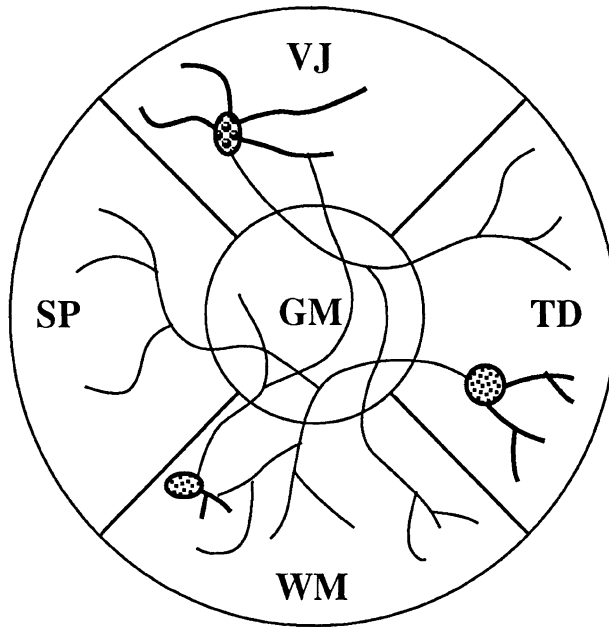


Figure 3. Neuronal pathways from a neuron in the TD module to one in the WM module to one in the VJ module and back to the TD module. Axon collaterals also go from the TD neuron to the SP module, and from the VJ neuron to the WM module. All pathways go through the GM module.

ORGANIZATIONAL HIERARCHY

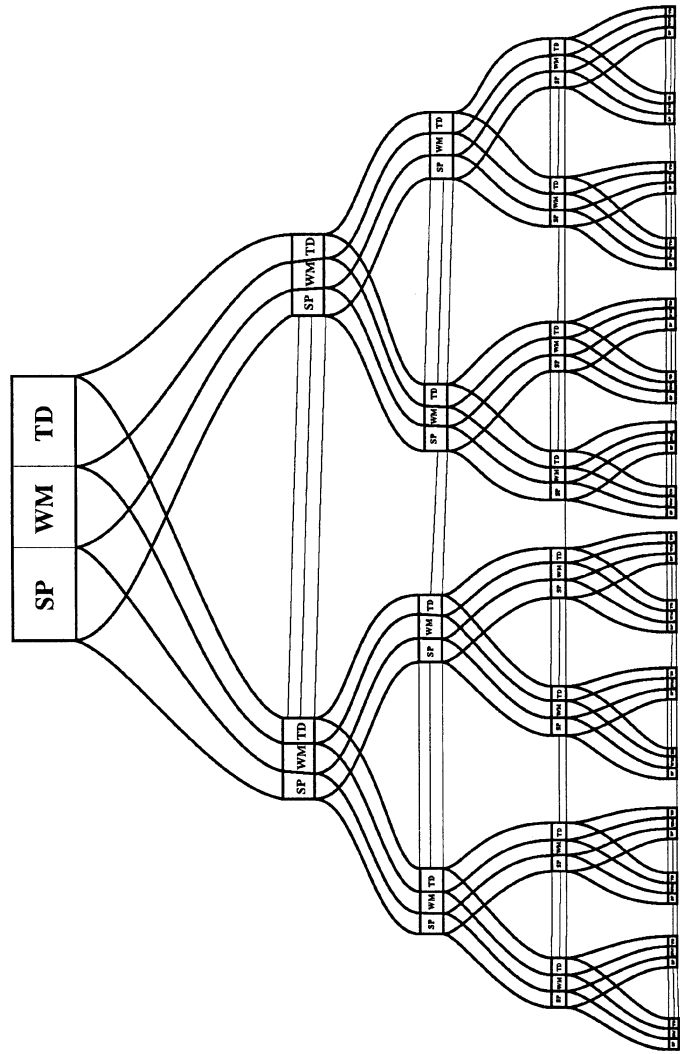


Figure 4. The organizational structure of RCS showing the spatial decomposition into subsystems and the horizontal communications between computing modules at the same level.

COMPUTATIONAL HIERARCHY

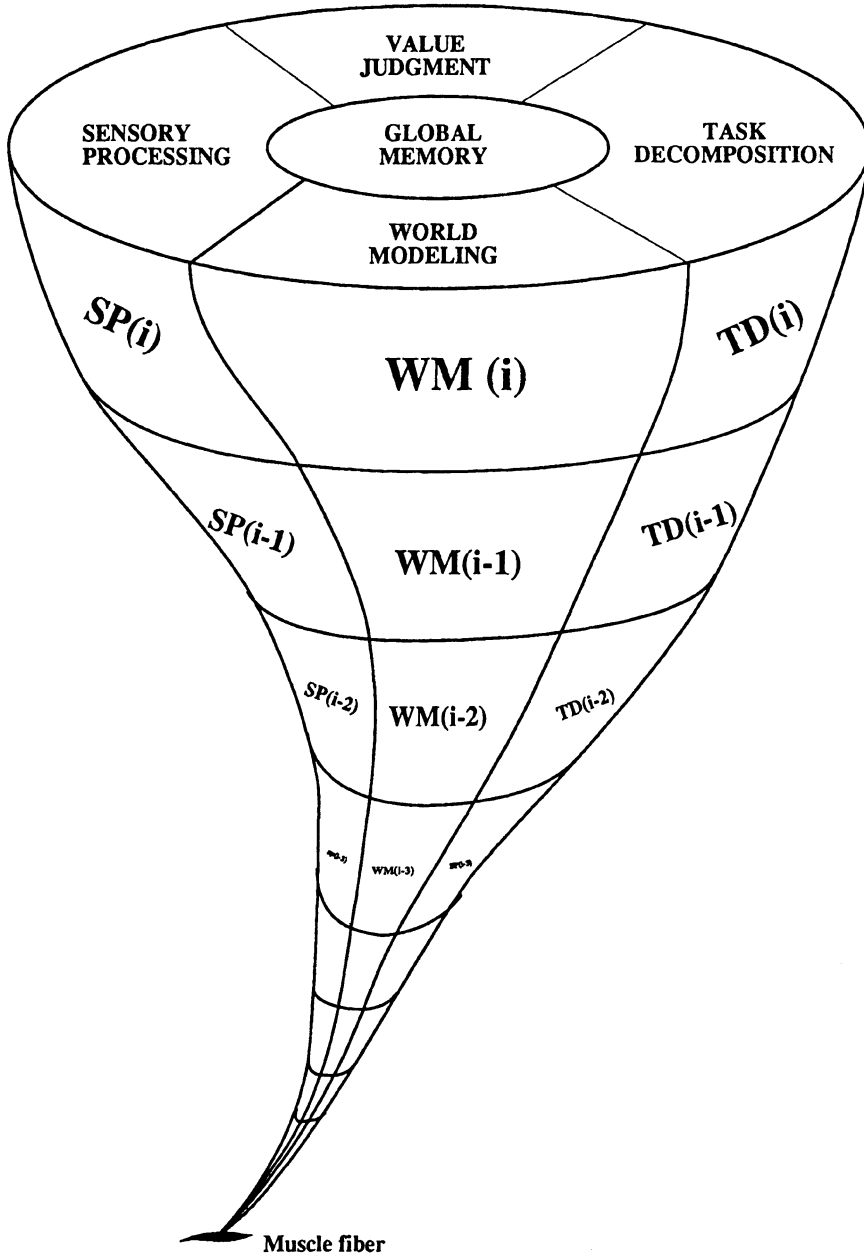


Figure 5. A single branch of the hierarchical tree of Figure 4, seen from a high level looking down to a single motor unit.

The JA, PL, and EX submodules in the task decomposition TD module communicate voluminously with each other and with the world modeling module at the same level. They negotiate with each other and resolve constraints. They ask What If? questions of the WM for planning and What Is? questions for execution.

WM modules are constantly in communication with SP modules at their corresponding level, predicting sensory input, and being updated by the observed state of the world. WM modules also respond to "What is?" and "What if?" questions from the executors and planners in the TD modules at their corresponding level.

Thus, although the proposed architecture incorporates a command and control hierarchy in the form of a formal logical tree, there exists an extensive horizontal flow of non-command information between TD, WM, SP, and VJ modules at the same level. In fact, the volume of information flowing horizontally completely dwarfs the amount flowing vertically.

The horizontal flow of information is, however, largely confined within the same subtree. The requirement for communications is much less between modules in separate subtrees. This is indicated by the number of horizontal lines in Figure 4.

It should be emphasized that global memory is not a single physical memory. In RCS, global memory is distributed over a number of single board computers and memory cards, and in many cases, these are distributed among separate vehicles.

Task Decomposition

For any intelligent system, there exists a set of tasks which the system knows how to do. Each task in this set can be assigned a name. The task vocabulary is the set of tasknames assigned to the set of tasks the system is capable of performing.

Task knowledge is knowledge of how to perform a task; plus information as what tools, materials, time, resources, and conditions are required; and what are the expected costs, benefits, and risks. A task frame is a data structure in which task knowledge is stored. For each taskname in the task vocabulary, there exists a task frame, or data structure, of the form:

TASKNAME	-- name of the task
a) actor	-- agent performing the task
b) action	-- activity to be performed
c) object	-- thing to be acted upon
d) goal	-- event that successfully terminates the task

- | | |
|-----------------|---|
| e) requirements | -- tools, time, resources, and materials needed |
| | -- conditions that must be satisfied to begin |
| | -- information that may be required |
| f) procedures | -- a state graph defining the plan (or schema) |
| | -- functions that may be called |
| | -- algorithms that may be needed |
| g) effects | -- expected results of task execution |
| | -- expected costs, risks, benefits |
| | -- estimated time to complete |

Task frames are essential to task planning. They are used by the task planners for generating hypothesized actions. They are used by the world model in predicting the results of hypothesized actions.

Task knowledge is typically difficult to discover, but once known, can be readily transferred to others. Task knowledge may be acquired by trial and error learning, but more often it is acquired from a teacher, or from written or programmed instructions. In most cases, the ability to successfully plan complex tasks is more dependent on the amount of task knowledge stored in the task frame than on the sophistication of a planner in reasoning about tasks.

For example, the common household task of preparing a food dish is typically performed by following a recipe. A recipe is an informal task frame for cooking. Gourmet dishes rarely result from reasoning about possible combinations of ingredients, still less from trial and error combinations of food stuffs. Recipes often are closely guarded secrets that, once published, can easily be understood and followed by others.

Task Decomposition (TD) Modules

TD modules in the task decomposition hierarchy plan and execute the decomposition of high level goals into low level actions. Task decomposition involves both a spatial decomposition (into concurrent actions by different subsystems), and a temporal decomposition (into sequential actions along the time line).

Each TD module at each level consists of three sublevels as shown in Figure 6. These three sublevels function to decompose the input task into both spatially and temporally distinct subtasks.

- 1) the job assignment sublevel -- JA submodule

The JA submodule is responsible for spatial task decomposition. It partitions the input task

Task Decomposition

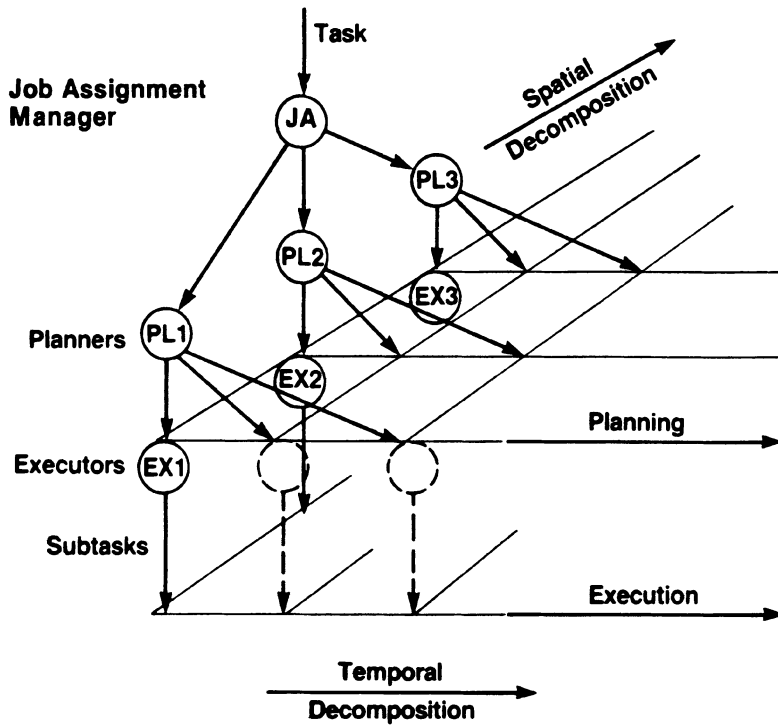


Figure 6. The job assignment JA module performs a spatial decomposition of the task. The planners PL(j) perform temporal decomposition. Each executor EX(j) closes a real-time control loop that serves the subtask to the plan.

command into N spatially distinct jobs to be performed by N physically distinct subsystems, where N is the number of subsystems controlled by the TD module.

The JA submodule is also responsible for assigning tools and allocating physical resources (such as arms, hands, legs, sensors, tools, and materials) to each of its subordinate subsystems for their use in performing their assigned jobs. These assignments are not necessarily static. For example, the job assignment submodule at the individual level may assign the arms to the manipulation subsystem in response to a <use tool> task command, and later assign the arms to the locomotion subsystem in response to a <swim> task command.

The job assignment submodule also selects the coordinate system in which the task is to be performed.

2) the planner sublevel -- planners $PL(j)$ $j = 1, 2, \dots, N$

For each of the N subsystems, there exists a planner submodule $PL(j)$. Each planner submodule is responsible for decomposing the job assigned to its subsystem into a temporal sequence of planned subtasks. Task planning typically requires evaluation of alternative hypothetical sequences of subtasks. Each planner $PL(j)$ functions by hypothesizing some action or series of actions. The WM module then predicts the effects of the action(s), and the VJ module computes the value of the resulting expected states of the world, i.e. a VJ evaluation function performs a priority-weighted cost-benefit analysis on the results predicted by the WM module. The evaluation values returned to the $PL(j)$ planner are sorted to select the hypothetical sequence of actions producing the best evaluation. This becomes the plan to be executed by $EX(j)$. This sequence corresponds to the communication flow depicted in Figure 3.

A job command to a subsystem may contain constraints on time, or specify job-start and job-goal events. A job assigned to one subsystem may also require synchronization with other jobs assigned to different subsystems. These constraints and requirements are specified by, or derived from, the task frame. Each planner $PL(j)$ submodule is responsible for checking its plan against plans generated by each of the other $N-1$ planners at the same level to determine if there are mutually conflicting constraints. If a conflict is found, constraint relaxation algorithms [27] may be applied until a solution is found. If no solution can be found, the planner reports failure to the job assignment submodule, and a new job assignment may be tried.

It must be noted that planning does not necessarily require reasoning from first principles as often presumed by AI planners. Planning can be accomplished by selecting from a library of partially or completely prefabricated plans [28], scripts [29], or schema [30].

For example, nature has provided many biological creatures with an extensive library of genetically determined plans, sometimes known as instinct. The range of behavior that can be generated by a library of instinctive plans at each hierarchical level, with each plan containing a number of conditional branches and error recovery routines, can be extremely large and complex. For many species, this library of plans is adequate to assure the survival of the species.

3) an executor sublevel -- executor submodules EX(j).

There is an executor EX(j) for each planner PL(j). The executor submodules are responsible for successfully executing the plans prepared by their respective planners. When an executor is informed by the world model that a subtask in its current plan is successfully completed, the executor steps to the next subtask in that plan. When all the subtasks in the current plan are successfully executed, the executor steps to the first subtask in the next plan. If the feedback indicates the failure of a planned subtask, the executor branches immediately to a preplanned emergency subtask. Its planner simultaneously begins work selecting or generating an error recovery sequence which can be substituted for the former plan which failed. Output subcommands produced by executors at level i become input commands to job assignment submodules in TD modules at level i-1.

Planners PL(j) constantly operate in the future, each generating a plan to the end of its planning horizon. The executors EX(j) always operate in the present, at time $t=0$, constantly monitoring the current state of the world reported by feedback F from the world model. In a sampled data control system, the executor modules operate on short, regular intervals, or execution cycles. The length of the execution cycle is set by a system state clock. The period of the state clock is defined by the rate at which sensory input data is sampled. At each state-clock cycle, each executor submodule compares the current step in its plan against the current state of the world, and computes an output subcommand designed to null the difference. The executors at each level react to feedback in one state clock period.

Thus, at each level, each executor submodule closes a reflex arc, or servo loop, and the executor submodules at the various hierarchical levels form a set of nested servo loops. In order to prevent instability due to nested servo loop interaction, the executor loop bandwidth decreases about an order of magnitude at each higher level.

Sensory Processing - SP modules

Sensory processing SP modules filter, correlate, and integrate sensory information over both space and time so as to detect, recognize, and measure patterns, features, objects, events, and relationships in the external world. This is shown in Figure 7. The SP modules consist of four

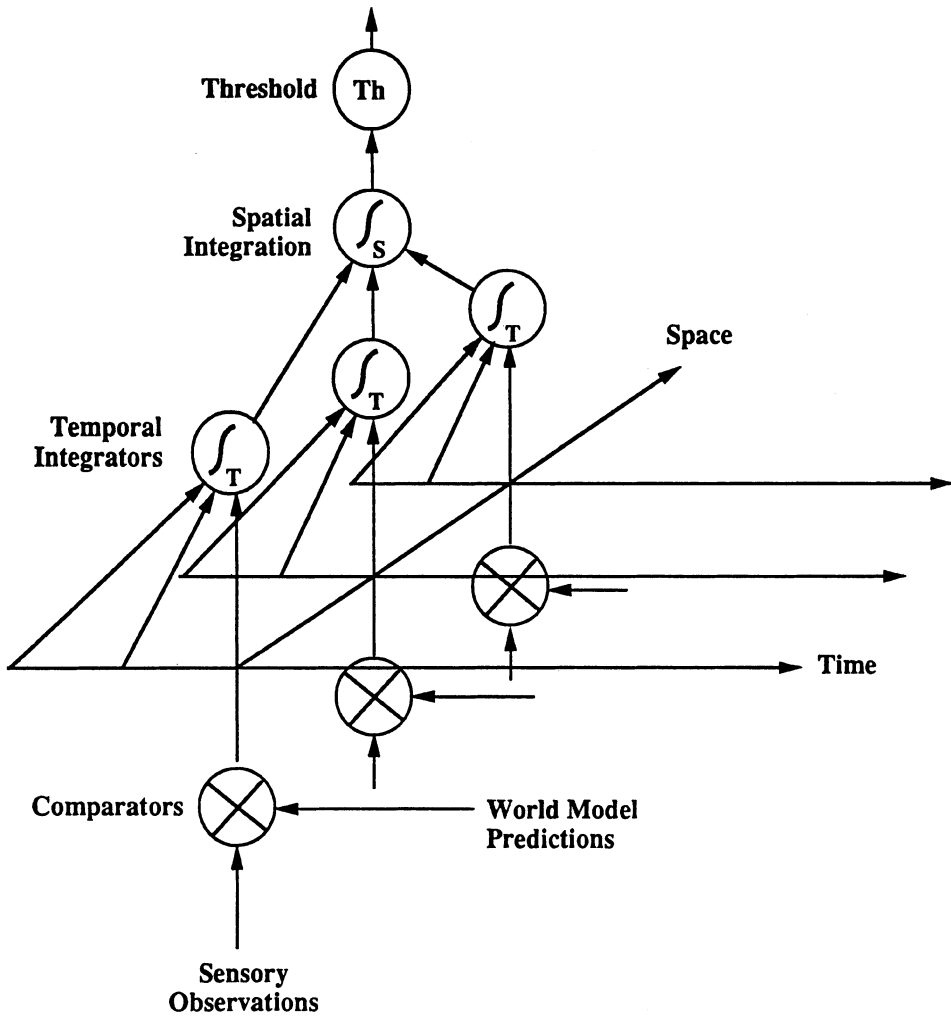


Figure 7. Each sensory processing SP module consists of: 1) a set of comparators that compare sensory observations with world model predictions, 2) a set of temporal integrators that integrate similarities and differences, 3) a spatial integration that fuses information from different sensory data streams, and 4) a threshold detector that recognizes entities and detects events.

sublevels:

- 1) A set of comparator modules
that compare sensor observations with world model predictions
- 2) A set of temporal integrators
that integrate similarities and differences over time
- 3) A set of spatial integrators
that integrate similarities and differences over space
- 4) A threshold module
that recognizes objects and detects events

The SP modules are dual to the TD modules. Whereas the TD modules decompose tasks spatially and temporally into subtasks for multiple subsystems, SP modules integrate data from multiple sources over extended time intervals into unified patterns, or gestalt perceptions of the world.

Similarities, differences, and newly detected or recognized events, objects, and relationships are entered by the WM modules into the world model global memory database. Objects or relationships perceived to no longer exist are removed.

World Modeling - WM modules

The world model is the system's internal best estimate of the history, current state, and possible future states of the world, including the states of the system being controlled. The world model includes both the WM modules and a knowledge base stored in global memory.

As shown in Figure 8, the WM modules at each level perform the following functions:

- a) Answer "What if?" questions asked by the job assignment and planner submodules in the corresponding level TD modules.
- b) Answer "What is?" questions asked by the JA, PL, and EX submodules in the corresponding level TD modules. The job assignment and planner submodules use this information for planning. The executors use it to monitor and servo the task.
- c) Provide predictions of expected sensory input to the corresponding SP modules, based on the state of the task and estimates of the external world.

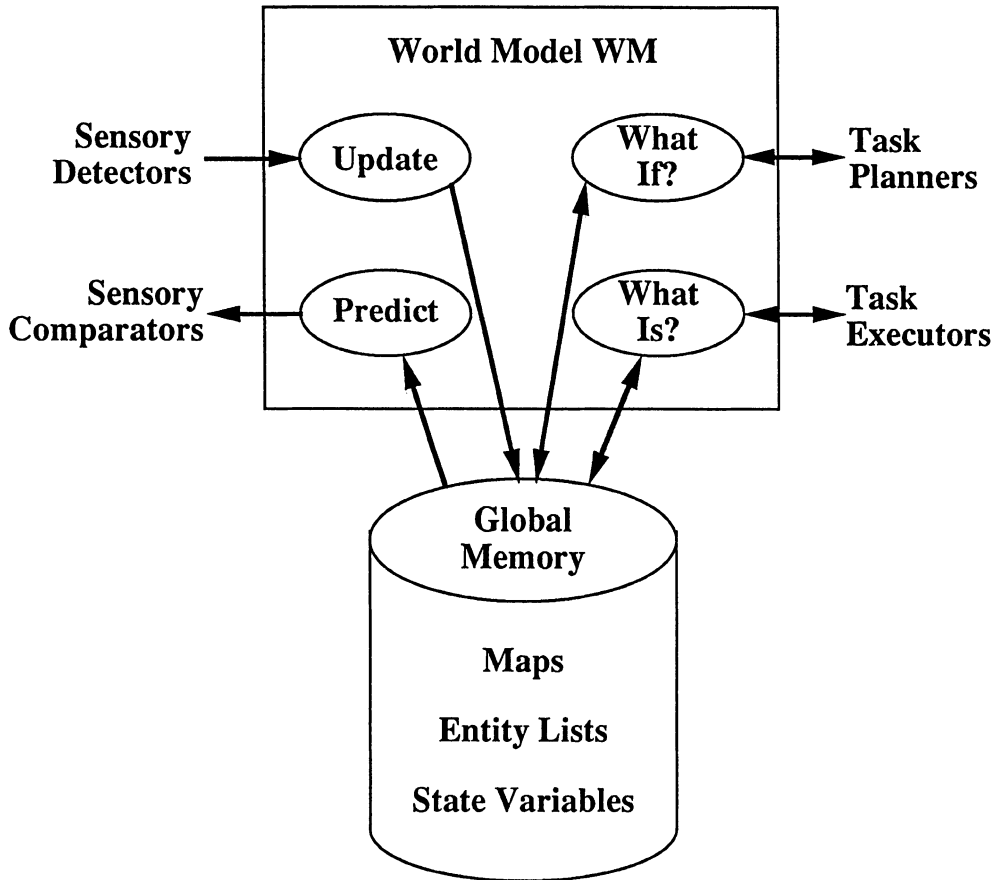


Figure 8. At each level, the world model responds to What If? and What Is? queries from task planners and executors, updates the global memory, and predicts sensory input.

d) Maintain the global memory knowledge base, keeping it current. The WM modules update the knowledge base based on correlations and differences between WM predictions and SP observations.

Contents of Global Memory

The knowledge in the global memory consists of maps, lists, and state variables.

a) Maps are 2-dimensional data structures that show the relative position of objects and regions. Maps may also contain overlays, that may indicate values such as utility, cost, risk, etc. assigned to regions or objects on the map. These values are computed by the VJ modules, and can be used both for planning and executing tasks.

Maps may be stored in two types of coordinate frames: world coordinates, and egosphere coordinates. These are illustrated in Figure 9.

A world coordinate map is a two dimensional representation in which latitude and longitude are typically the x-y coordinates, and each pixel contains a pointer to a data structure that gives the physical properties and z-dimension of the region or objects covered by that pixel. A vehicle moving through the world can be represented as an object moving on the world map. The world map may be scrolled so as to keep a particular vehicle of interest at the center.

An egosphere is a polar coordinate system map centered on the sensor system. Pixels are referenced by azimuth and elevation, and each pixel contains a pointer to a data structure that gives the physical properties and range of the region or object covered by that pixel. As the system moves through the world, the contents of the pixels on the egosphere change. Thus, the egosphere must be periodically recomputed to be current.

Each hierarchical level may use a different resolution working copy of a portion of the world map. The working copy at each level covers a region that bounds the task being planned at that level with resolution such that subtasks being planned at that level are easily resolved. Each level may also use a different set of egosphere overlays.

b) Entity lists

All entities of interest, such as objects, tasks, features, regions, and relationships, and events, are listed in the global memory database, indexed by name and characteristic features. Each entity in the list has a data form, or "frame", containing attributes. For example an object frame might have the form:

Hierarchical Planning

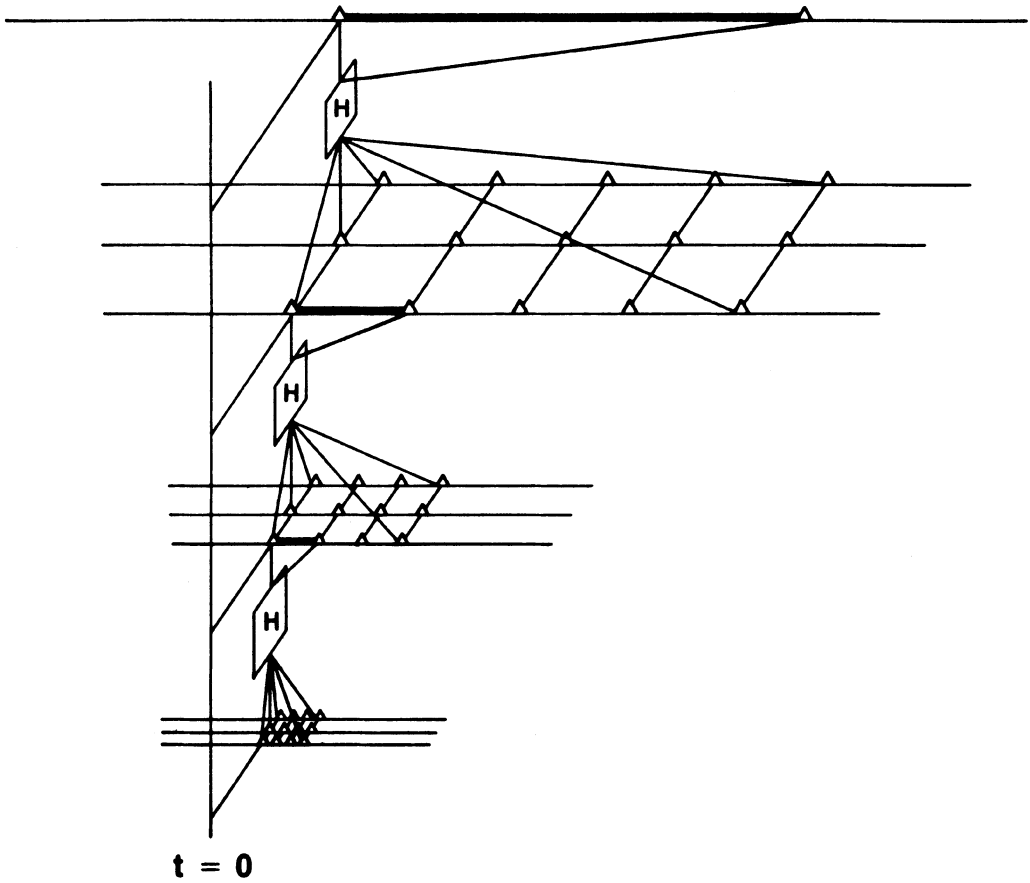


Figure 9. Geometric relationships between world map and egosphere coordinates.

ENTITY NAME	-- Name
Type	-- (Object, event, task, region, etc.) -- (Specific or generic)
Geometry	-- Shape -- Size -- Sub-entity list -- Parent entity
Position	-- World map coordinates -- Egosphere coordinates
Dynamics	-- Velocity
Linear	
Rotational	-- Acceleration
Attributes	-- Model or class -- Mass -- Color -- Substance -- Typical behavior (of intelligent entities) -- Speed -- Range
Value state variables	-- (Attract-repuls, etc.) -- Confidence levels

For moving objects, entity frames may contain not only current map coordinates, but a past history or trace of coordinate positions. Event frames contain information such as start and end time, duration, type, cost, payoff, etc. At different hierarchical levels, object frames have different levels of detail and spatial resolution, and event frames have different levels of temporal resolution.

c) State Variables

State variables may be entries in entity frames, or may define values for maps overlays. The entire set of state variables define the system's best estimate of the state of the world, including both the external environment and the internal state of the TD, WM, SP, and VJ modules.

Value Judgments -- VJ modules

The value judgment VJ modules contain functions that enable them to evaluate events, situations, or objects observed by sensory input, or stored in the world model, or predicted from the world model. VJ modules also contain functions which compute confidence factors and probabilities of recognized events, and statistical estimates of stochastic state variables.

Inputs to the VJ modules are state variables, goal priorities, and values assigned to entities such as vehicles, targets, and resources.

Outputs from the VJ modules are value state-variables, such as good, bad, cost, risk, payoff, attractive, repulsive, etc., that may be assigned to entity frames or map regions. For example, attraction or repulsion values may be assigned to objects, places, or persons. Values of cost, risk, and payoff may be assigned to events such as subtask completion, information acquisition, and vehicle survival. Cost and risk values may also be associated with map route segments.

VJ modules also compute subgoal priorities, and assign value state-variables such as cost and benefit to subtask goals for evaluating plans and choosing execution options. For example, in planning the planners hypothesize actions, and the WM modules generate predictions of what will result from hypothesized actions. The VJ modules apply evaluation functions to these results and return value state-variables to the planners. These values can then be assigned to nodes in a game tree as the planners conduct a search over the space of possible futures. The planners then choose the sequence of hypothesized actions that produce the best evaluation as the plan to be executed.

Executors may also use value state-variables from the VJ modules to select among various possible next actions in a plan state- graph. Value state-variables computed by the VJ modules on the current state of the world can be assigned to variables in the plan graph edge predicates, to be used by the executors to make moment by moment behavioral decisions.

In biological creatures, the VJ modules reside in the limbic system.

Timing

A timing diagram illustrating the temporal flow of activity in the task decomposition and sensory processing systems is shown in Figure 10. Temporal decomposition along the time axis produces a "dechunking" of tasks into subtasks according to their characteristic time intervals for execution. These characteristic time intervals provide the primary criteria for defining hierarchical levels in the control architecture.

For example,

At level 1, feedback inputs and command outputs occur on the order of every few milliseconds. Command inputs occur around every 30 milliseconds. Smooth motion is planned for each actuator, and servo loops control position, velocity, and force.

At level 2, feedback inputs and command outputs occur about every 30 milliseconds. Command

inputs occur around every 300 milliseconds. Dynamically coordinated motion is planned and controlled for tightly coupled groups of actuators.

At level 3, feedback inputs and command outputs occur about every 300 milliseconds. New command inputs occur around every 3 seconds. Obstacles are avoided and clear path trajectories are planned and executed for entire subsystems such as manipulation and locomotion.

At level 4, feedback inputs and command outputs occur about every 3 seconds. Command inputs change on average about every 30 seconds. Simple tasks on single entities are planned and sequenced for the whole individual.

At level 5, feedback inputs and command outputs occur about every 30 seconds. Command inputs change on average about every 5 minutes. Complex tasks by one or more actors on multiple entities are planned and sequenced for the self group. In each individual, level 5 defines that individual's role in the group.

At level 6, feedback inputs and command outputs occur about every 5 minutes. Command inputs change on average about every hour. Tasks by one or more groups on other groups are planned and executed. In each individual, level 6 defines that individual's role in intergroup activities.

In the temporal decomposition hierarchy, each higher level produces about an order of magnitude increase in planning horizon and decrease in plan resolution along the time line. The rate of subtask execution specifies the loop bandwidth requirements for the feedback control loop through each level. When combined with vehicle velocity, it also defines the extent of the spatial regions over which tasks are planned and executed. Thus, temporal decomposition produces not only a functional task decomposition, but a hierarchical layering of control loops and a hierarchical structuring of world model maps and entities that support the temporal and functional decompositions.

In the model presented here, input to the intergroup level is derived from a top level executor which operates on a top level plan. The top level plan may be determined by instinct and habits of daily routine, or it may be developed by a top level planner that generates or selects plans based on VJ evaluations. Command input to the top level planner consists of the single unchanging goal, <propagate genes>.

Feedback input to the top level executor consists of high level events and state variables such as hormone levels, circadian rhythms, drives of hunger, thirst, and sex, the recognition of objects and relationships between objects, and the detection of danger, or other emotionally stimulating events.

At each lower level, plans are formulated (or selected) in real- time to accomplish the current and next task in the plan of the level immediately above. Each task in the higher level plan is decomposed into a lower level plan of at least two, and on average, about 10 subtasks. The planning horizon thus shrinks exponentially by about an order of magnitude at each successively lower level of the hierarchy.

Figure 11 shows a simple example of three levels of planning activity represented in Gantt notation. The input command to the top level TD module is decomposed by the job assignment manager and three planners of the top TD module into three simultaneous plans consisting of four subtasks each. The first executor of the top level TD module outputs the current subtask command in its plan to a second level TD module. This second level task command is decomposed by the job assignment manager and three planners in the second level TD module into three plans, again consisting of four subtasks each. The first of the second level executors outputs the current activity in its plan to a third level TD module, which further decomposes it into three plans of four subtasks. At each level the final subgoal events in the plans correspond to the goal of the input task. At each successively lower level, the planning horizon becomes shorter, and the subtasks become more detailed and fine structured.

The timing ratio between levels is not at regular, particularly at the higher levels, because executor outputs are event driven, triggered by feedback from the environment. Depending on events in the world, the rate of change of subtasks may vary considerably. Furthermore, an output from any level may be repeated any number of times. Thus, the apparent interval between outputs may vary over an even wider range.

Figure 10 also illustrates the duality between the task decomposition and the sensory processing hierarchies. At each level in the hierarchy, the sensory processing modules look back into the past about as far as the planner modules look forward into the future. At each level, future plans have about the same detail as historical traces.

5. Cooperation Between Multiple Individuals

In a factory control system, where high bandwidth communication is always available between all control modules, there can be a single group control module for each group, and a single intergroup control module to coordinate the activities of the various groups.

In the case of autonomous individuals, however, communication may not always be available, or may be corrupted by noise, or be limited in bandwidth. In order to cope with these situations, each autonomous individual should contain its own group and intergroup control levels. This enables individuals to execute cooperative group and intergroup tasks with a minimum of communication.

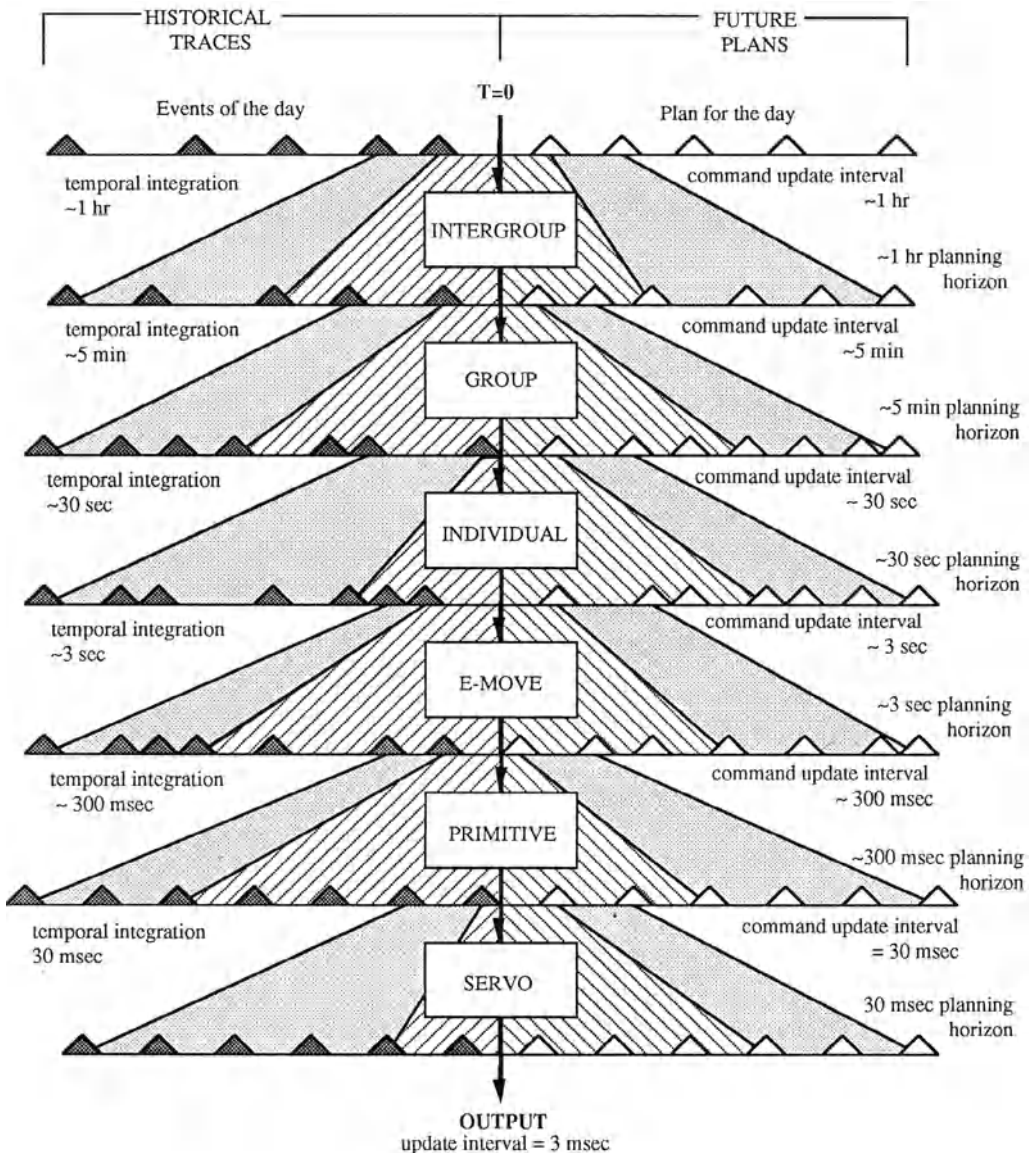


Figure 10. A timing diagram illustrating the temporal flow of activity in the task decomposition and sensory processing systems. Input to the intergroup level is derived from a top level executor where high level sensory events and circadian rhythms react with habits and daily routines that constitute the plan for the day.

If the level 5 and 6 TD modules of each individual contain the same algorithms and task vocabularies as the group and intergroup leaders of that individual, and those TD modules contain the same version of the command tree, and have the same information resident in their world models, and have the same set of goals, values, and priorities, then each individual can duplicate the computations performed by its group and intergroup leaders. Under such conditions, each individual can compute what commands it would receive if communications were available. It then can operate as if it were in constant communication, both with its superiors and peers, and cooperative behavior can take place without any communication at all. Even under conditions where task sequencing is event driven, if all relevant events can be sensed independently by all the cooperating individuals, task sequencing can be performed simultaneously by each individual, and communication is unnecessary.

Of course, such ideal conditions rarely obtain. Although, it is relatively simple and inexpensive to equip all individuals with the level 5 and 6 computing structure and algorithms, identical goals, priorities, values, and knowledge can seldom be long maintained for multiple independent individuals, and all relevant events will seldom be sensed and interpreted identically by all individuals. Communication thus becomes necessary for cooperative behavior. The role of communication is:

- 1) To issue commands and achieve a common goal
- 2) To define tasks and achieve a common strategy
- 3) To establish roles and achieve a common command tree
- 4) To communicate world states and achieve common world models.

The amount and frequency of communication required to achieve successful cooperative behavior is then inversely proportional to the amount of commonality in goals, priorities, values, and knowledge.

REFERENCES

- [1] Decker, K.S., "Distributed Problem-Solving Techniques: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, Volume SMC-17, No. 5, September/October 1987.
- [2] Erman, L.D., Hayes-Roth, F., Lesser, V.R., and Reddy, D.R., "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty," Computer Surveys, Volume 23, pp. 213-253, June 1980.
- [3] Fox, M.S., "An Organizational View of Distributed Systems," IEEE Transactions on Systems, Man, and Cybernetics, Volume SMC-11, pp. 70-80, January 1981.
- [4] McLean, C.R. and Brown, P.E., "Process Planning in the AMRF," CIM Technology Magazine, August 1987.
- [5] Cammarata, S., McArthur, D., and Steeb, R., "Strategies of Cooperation in Distributed Problem Solving," Proceedings 8th International Joint Conference on Artificial Intelligence, pp. 767-770, August 1983.
- [6] Thorndyke, P.W., McArthur, D., and Cammarata, S., "Autopilot: A Distributed Planner for Air Fleet Control," Proceedings 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, pp. 171-177, August 1981.
- [7] Albus, J.S., "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles," NIST Technical Note 1251, September 1988.
- [8] Nilsson, N.J., "Two Heads are Better than One," SIGART Newsletter, p.43, October 1980.
- [9] Minsky, M., Society of Mind, Simon and Schuster, New York, 1986.
- [10] Lard, J.E., Newell, A., and Rosenbloom, P.S., "SOAR: An Architecture for General Intelligence," Artificial Intelligence 33 (1987), pp. 1-64, Elsevier Science Publishers B.V. (North Holland).
- [11] Intelligent Task Automation Interim Technical Report, Report No. II-4, Honeywell Corporation Systems, Development Division, December 1987.
- [12] Lowrie, J., et al., "Autonomous Land Vehicle," Annual Report, ETL-0413, Martin Marietta Denver Aerospace, July 1986.
- [13] Albus, J.S., Barbera, A.J., and Nagel, R.N., "Theory and Practice of Hierarchical Control," Proceedings of the 23rd IEEE Computer Society International Conference, September 1981.
- [14] Barbera, A.J., Albus, J.S., Fitzgerald, M.L., and Haynes, L.S., "RCS: The NBS Real-Time Control System," Robots 8 Conference and Exposition, Detroit, MI, June, 1984.
- [15] Leake, S. and Kilmer, R.D., "The NBS Real-Time Control System User's Reference Manual," NBS Technical Note 1258, June 1988.

- [16] Shneier, M., Kent, E.W., Albus, J.S., Mansbach, P., Nashman, M., Palombo, M., Rutkowski, W., and Wheatley, T., "Robot Sensing for a Hierarchical Control System," Proceedings of the 13th ISIR/Robots 7 Symposium, Chicago, IL, April 1983.
- [17] Kent, E.W. and Albus, J.S., "Servoed World Models as Interfaces between Robot Control Systems and Sensory Data," *Robotica* (1984) volume 2, pp. 17-25.
- [18] Fiala J. and Wavering A., "An RCS Application Example: Tool Change on a Horizontal Machining Center," Proceedings of the Second International Conference on Robotics and Factories of the Future, San Diego, CA, July 28-31, 1987.
- [19] Murphy, K.N., Norcross, R.J., and Proctor, F.M., "CAD Directed Robotic Deburring," Proceedings of the Second International Symposium on Robotics and Manufacturing Research, Education, and Applications, Albuquerque, NM, November 1988.
- [20] McLean, C.R., "The Vertical Machining Workstation of the AMRF: Software Integration," Proceedings of the ASME Symposium on Integrated and Intelligent Manufacturing, Anaheim, CA, December 1986.
- [21] Kramer, T.R., "Data Handling in the Vertical Workstation of the Automated Manufacturing Research Facility at the National Bureau of Standards," NSBIR 88-3763 (NB), April 1988.
- [22] Hopp, T.H., "CAD-Directed Inspection," *CIRP Annals*, Volume 33/1, 1984.
- [23] Mitchell, M.J. and Barkmeyer, E.J., "Data Distributed in the NBS Automated Manufacturing Research Facility," Proceedings of the IPAD2 Conference, Denver, CO, April 17-18, 1984.
- [24] Furlani, C. (Editor), "Hierarchical Control System Emulation User's Manual," NBSIR 85-3156, April 1985.
- [25] McCain, H.G., Kilmer, R.D., Szabo, S., and Abrishamian, A., "A Hierarchically Controlled Autonomous Robot for Heavy Payload Military Field Applications," Proceedings of the International Congress on Intelligent Autonomous Systems, Amsterdam, The Netherlands, December 8-11, 1986.
- [26] Albus, J.S., McCain, H.G., and Lumia, R., "NASA/ NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)," NBS Technical Note 1235, July 1987.
- [27] Fikes, R.E. and Nilsson, N.J., "Strips: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence*, 2, (2), pp. 189-208, 1971.
- [28] Sacerdoti, E., A Structure for Plans and Behavior, New York Elsevier North Holland, 1977.
- [29] Schank, R.C. and Abelson, R.P., Scripts Plans Goals and Understanding, Hillsdale, NJ, Lawrence Erlbaum Associates, 1977.
- [30] Arbib, M.A., The Metaphorical Brain, Wiley, New York, 1972.

Cellular Robotics

Construction of Complicated Systems from Simple Functions

Toshio Fukuda * Yoshio Kawauchi**

*Nagoya Univ. Dept. of Mechano-Informatics

Furo-cho, Chikusa-ku, Nagoya 464-01, Japan

**Toyo Engineering Corp.

2-8-1 Akanehama, Narashino-shi, Chiba 275, Japan

Abstract

This paper deals with a method of constructions complicated systems based on cells having simple functions. The analogy with the human body is obvious. A "cell" in this paper is defined as a functional element used to build a robotic system from a biometrics viewpoint. Cells can dynamically reconfigure the system depending on given tasks. Complicated robotic systems can be made from autonomous cells have the capacities of sensing, decision making, and self-control.

This paper shows the cellular robotics, called "CEBOT", in the following items:

- (1) Mechanical structure of cells,
- (2) Sensing and obstacle avoidance systems,
- (3) Rendezvous and docking systems,
- (4) Control System of overall architecture,
- (5) Communications system for recognition and identification,
- (6) Optimal manipulator configuration method,
- (7) Knowledge-base software architecture, and
- (8) Task planning systems.

Demonstrations can be given using mobile and other cells, so that cells act together to build more complicated systems. In the process of

constructions a more complicated system, messages from one cell must be transferred to other cells through communication channels before and after docking. In CEBOT, optical communication is employed before docking occurs. After docking, the bus architecture of the cell is used to communicate with all connected cells. Finally, many useful aspects of this system are pointed out for robotic applications, such as in spacearea and other industrial fields.

1. Introduction

The novel concept of **Dynamically Reconfigurable Robotic System (DRRS)** has some very unique features. The fundamental premise idea is similar to that of biological organisms which consist of basic elementary cells having simple functions. If many fundamental cells conglomerate to organs, the organs can perform complicated functions. DRRS uses the same principle as elementary functional cells, which have a simple mechanical function and a coupling mechanism to connect with other cells. By connecting many cells to modules or large structures, very sophisticated tasks can be executed. DRRS system components are widely distributed, autonomous and freely exchangeable, which leads to fault-tolerance and self-repair capabilities. These are very unique features not available in conventional robotic systems. The flexible DRRS is considered to be an optimal robotic system for many present and future applications.

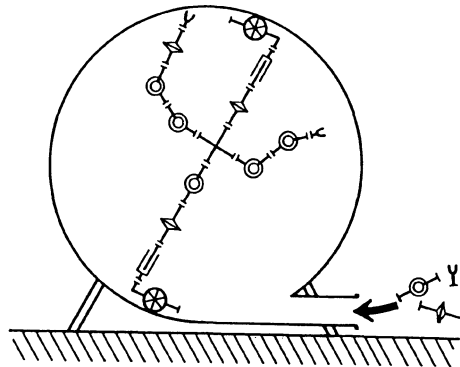


Figure 1: Application of DRRS inside a tank

An example of maintenance work inside a tank is shown in Fig.1. Single components of DRRS are fed through the narrow tank inlet, can build up a large structure which can maintain the tank very efficiently.

The hardware realization of the new DRRS concept as **Cellular Robotic System CEBOT** is shown in Fig. 2. This series I CEBOT consists of mobile cells, bending joint cells rotating joint cells, and end-effector cells [1]. The series II CEBOT (see Fig. 3, Fig. 4) has a new coupling mechanism (hook-type)[2]. The cone-shaped design guides the cells during automatic approaches, thus improving coupling characteristics.

A description of the cell and coupling mechanism design, automatic approach, docking, and structure planning methods for CEBOT can be found in Sect. 2. Section 3 is concerned with the communications system for the undocked and docked state of CEBOT. In Sect. 4 the concept of an universally applicable end-effector structure planning expert system is described.

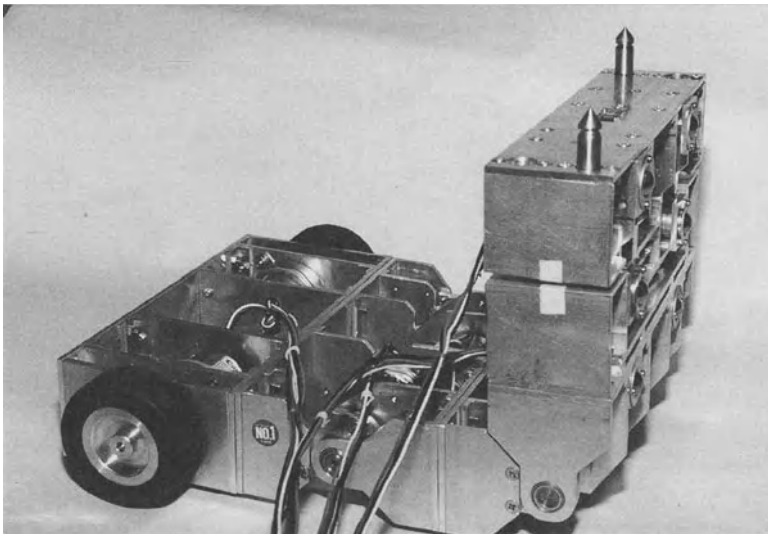


Figure 2: Photo of Cellular Robotic System CEBOT, series I

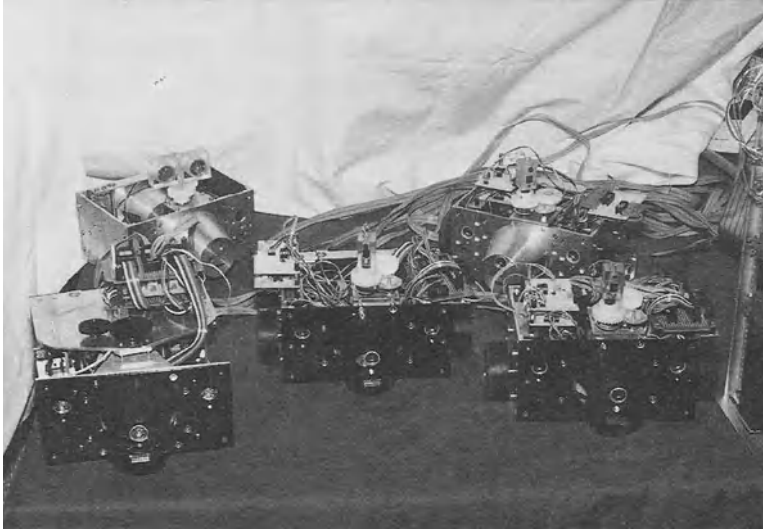


Figure 3: Photo of Cellular Robotic System CEBOT, series II (undocked state)

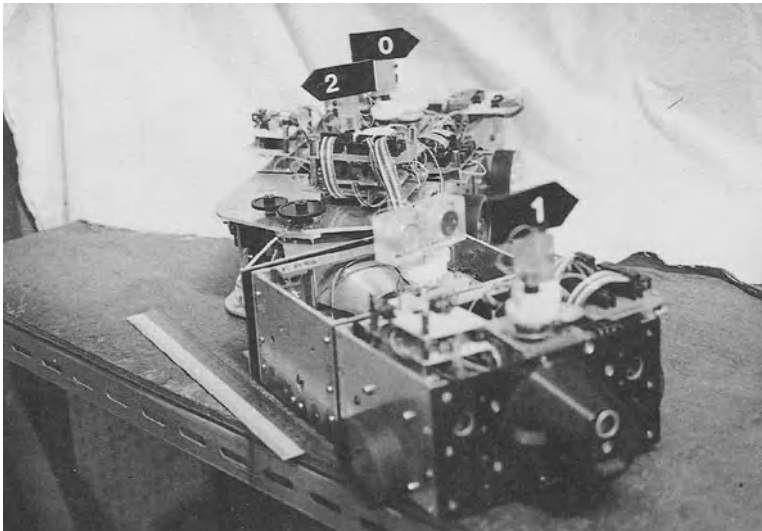


Figure 4: Photo of Cellular Robotic System CEBOT, series II(docked state).

2. Concept

2.1 Concept of Dynamically Reconfigurable Robotic Systems (DRRS)

DRRS uses an approach similar to that of biological systems. The idea is to have a few, very simple components, which are able to connect and separate freely. In this way a flexible dynamically changeable structure is realizable. Many cells can connect to modules, and many modules can interact to create very complex systems. DRRS has many unique features similar to these of biological organisms. Self-redundancy on the cell/level can be achieved by replacing defective components or by repairing faults using end-effector cells included in the entire DRRS. If repair or replacement is not possible, other system modules or components can substitute for the function of defective parts. The flexible hardware structure of DRRS, as well as its software system, are freely re-configurable.

A single component of DRRS, referred to as a "cell", has its own intelligence, self-contained software, independent data base, sensor system, communication capabilities, a coupling mechanism, and a single mechanical function. Cells are classified in 3 groups according to their mechanical function:

Group 1: Active actuating cells such as; bending joint cells, rotating joint cells, sliding joint cells, end-effector cells, and mobile cells (wheel, crawler, etc.).

Group 2: Branching cells, orientation changing cells, length adjustable arm cells between joints, power cells for heavy duty works, etc.

Group 3: Work cells, end-effector cells, and special purpose cells.

Cell connection and separations are carried out automatically by mobile cells interacting with other cells. Joint and end-effector cells, which are already combined to form a manipulator, can attach or detach other cells.

Like biological organisms, DRRS is also organized in a hierarchically structured system (see Fig.5, Fig.6). In the lowest level (LEVEL 5), we find cells with the basic function of group 2,3. These cells have their own independent CPU, self-containing software, data base, and sensor systems. They are principally completely autonomous intelligent units. However, basic cells of LEVEL 5 are not able to move actively, which distinguishes them from the mobile or joint cells of group 1. These active cells are placed in LEVEL 4. Cells of LEVEL 4 and 5 can freely communicate with each other. Cells in LEVEL 4 are masters of modules, which they form together with several LEVEL 5 cells. When building up modules or larger structures, the module master searches for its partner cells and automatically approaches and docks with them.

Several modules can combine to form large structures, a process which is controlled by the coordination level (LEVEL 3). In many applications the coordination of uncoupled modules or mobile cells is desirable, and is also controlled in LEVEL 3.

The structure planning level (LEVEL 2) generates structures depending on given tasks. The required number of cells, the cell types, and their combination sequences are decided.

Input commands to the structure planning level arise from the task planning level (LEVEL 1), which is the highest level, and corresponds to the central nervous system of biological organisms.

2.2 Realization of DRRS as the Cellular Robotic System CEBOT

2.2.1 Hardware of CEBOT

The hardware realization of the new DRRS concept as **Cellular Robotic System CEBOT** was shown in Fig. 2. This series I CEBOT consists of a mobile cell, a bending joint cell, a rotating joint cell and an end-effector cell. The SMA-spring actuated coupling mechanism is shown in Fig. 7. Automatic approaches must be performed due to the small guiding pins which only slightly guide the automatic docking procedure. The connectable area ratio (i.e. the ratio of the area from where docking can occur and the surface area of the cell front.) was 0.3%.

CEBOT series II cells (see Fig. 3) have a new coupling mechanism(hook-type). The cone-shaped design guides the cells during approach and the hook fixes the connection (see Fig. 8). The connectable area ratio has been improved to 4%, which allows more uncertainty during automatic approach.

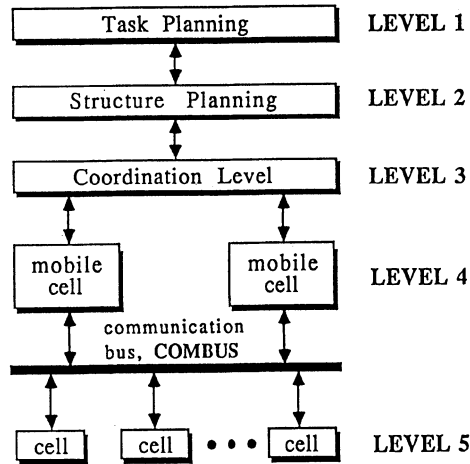


Fig. 5: Hierarchical system structure of DRRS

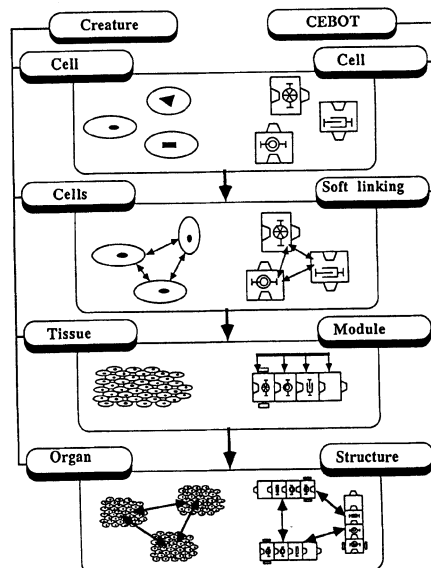


Fig. 6: Analogy of CEBOT and biological systems

2.2 Realization of DRRS as the Cellular Robotic System CEBOT

2.2.1 Hardware of CEBOT

The hardware realization of the new DRRS concept as **Cellular Robotic System CEBOT** was shown in Fig. 2. This series I CEBOT consists of a mobile cell, a bending joint cell, a rotating joint cell and an end-effector cell. The SMA-spring actuated coupling mechanism is shown in Fig. 7. Automatic approaches must be performed due to the small guiding pins which only slightly guide the automatic docking procedure. The connectable area ratio (i.e. the ratio of the area from where docking can occur and the surface area of the cell front.) was 0.3%.

CEBOT series II cells (see Fig. 3) have a new coupling mechanism(hook-type). The cone-shaped design guides the cells during approach and the hook fixes the connection (see Fig. 8). The connectable area ratio has been improved to 4%, which allows more uncertainty during automatic approach.

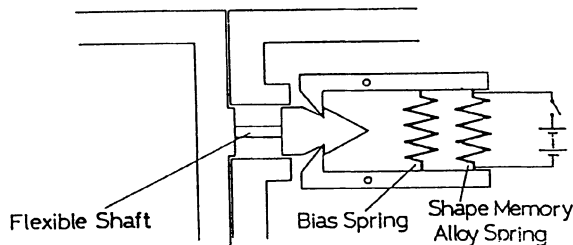


Fig. 7: SMA actuated pin type coupling mechanism, series I

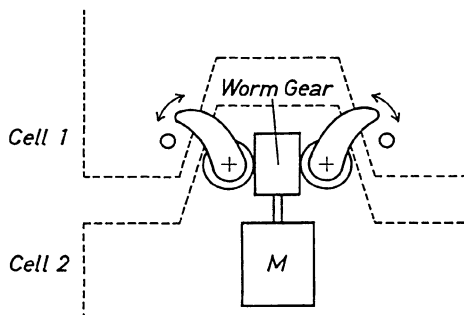


Fig. 8: Hook-type coupling mechanism, series II CEBOT

2.2.2 Sensor System

The sensor system of CEBOT has sensors for cell recognition (infrared), position and attitude detection (infrared), and obstacle avoidance (ultrasonic).

For short distance position and attitude location (0-0.5m), 8 infrared LEDs in the object and 3 infrared photodiodes in the mobile cell are used. The 8 LEDs are facing 8 different directions at 45° intervals (see Fig. 9). One ultrasonic transmitter and 2 receivers are used for obstacle detection (see Fig. 10). This sensor combination can recognize obstacles as far as 0.7 m away and at inclination angles of up to 30° .

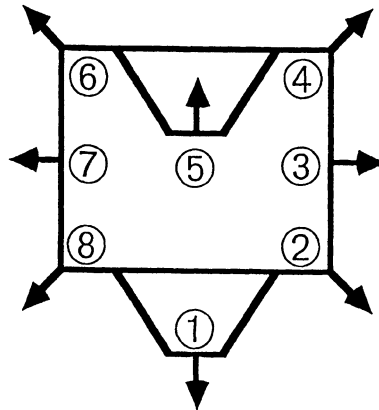


Fig. 9: Eight LEDs for position and attitude detection

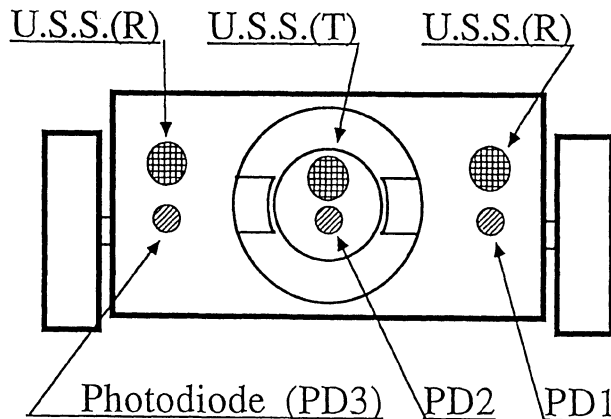


Fig. 10: Photodiode and ultrasonic sensors

Figure 11 shows an experimental setup with an object cell in the center and a moving cell 0.4m on its circumference. Depending on angle q , the center photodiode PD2 of the moving cell gives different outputs for different LEDs of the object cell (see Fig.12).

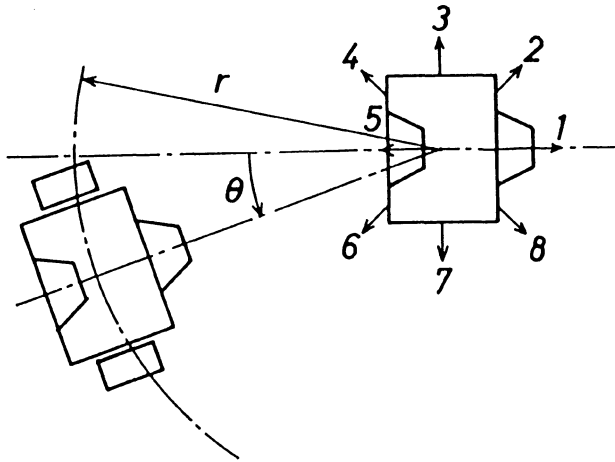


Fig. 11: Experimental setup for object cell attitude recognition

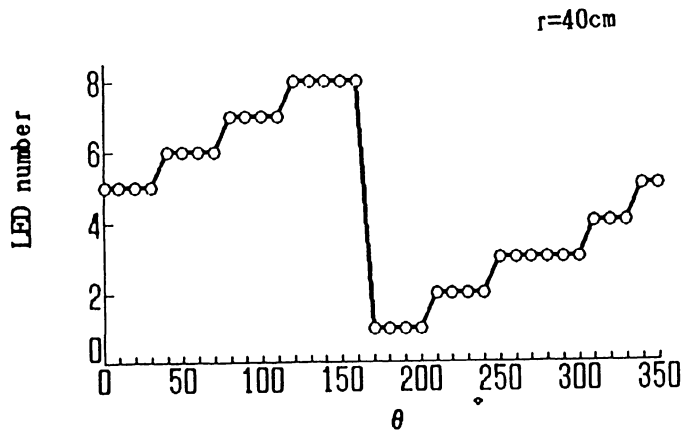


Fig. 12: LED number dependent on angle θ

2.2.3 Automatic Approach and Docking Method

Automatic approach of CEBOT cells is done by a 5-step algorithm, which is separated in rough-approach (STEP 1,2), attitude finding (STEP 3), and fine-approach (STEP 4 and 5) phases.

STEP 1

The mobile cell selects the desired cell type out of all possible object cells. It then rotates around its axis, recognizing the object cell position from the output of photodiode PD2 (see Fig. 13).

STEP 2

The mobile cell approaches the object until a minimum distance is reached. The minimum distance is defined as the output level of PD2 exceeding a given threshold. During this rough approach the movement direction is controlled by minimizing the output difference between photodiodes PD1 and PD3 (see Fig. 13).

STEP 3

The object cell activates one LED after another so that the mobile cell can detect its attitude. It then moves along a default path near the back side of the object cell (see Fig. 14).

STEP 4

The mobile cell makes a fine approach to the connectable area from where coupling can be executed (see Fig. 15).

STEP 5

Final connection is made by closing the hook-type coupling mechanism (see Fig. 15).

An experimental result of an automatic approach is shown in Fig. 16.

This 5-step approach algorithm has been refined to fit the new implementation in this work. It now includes object cell type recognition by communication in undocked state before step 1. Object cell distance, position and relative attitude can now be measured by a rotating infrared sensor, which substitutes for steps 1,2,3 and makes automatic approach

more efficient. The new implementation also allows sensing and moving of mobile cells. For more implementation details see Sect. 4.

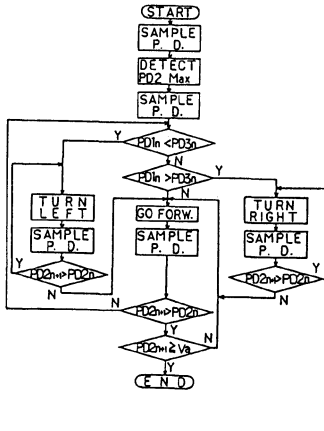


Fig. 13: Approach algorithm,
rough approach, STEP 1,2

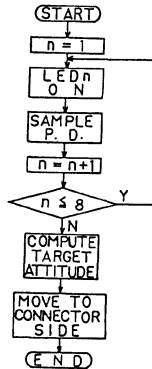


Fig. 14: Approach algorithm, object attitude finding, STEP 3

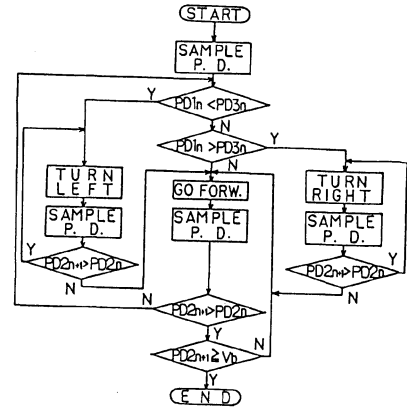


Fig. 15: Approach algorithm,
fine approach, STEP 4,5

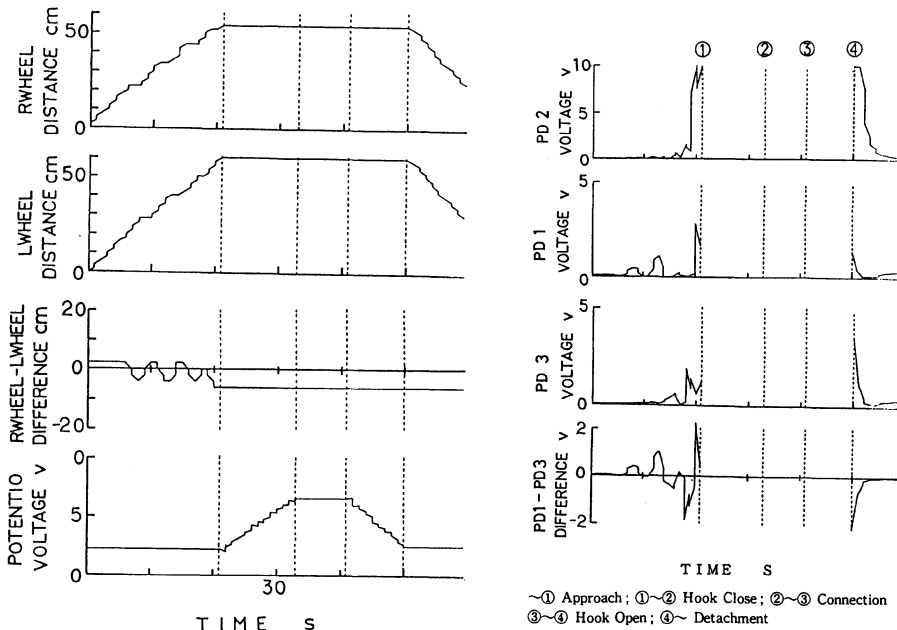


Fig. 16: Experimental result of automatic approach

2.2.4 Structure Planning Method

To automatically generate a CEBOT structure for a given task, a simple 3 degree of freedom manipulator approach leads to a table of candidates for given task work points. These candidates have to fulfill three geometric constraints, i.e. link interference, maximum joint movement limits and combination infeasibility. Common candidates, which are applicable for all work points are selected and evaluated by an evaluation function. This evaluation function is adaptable to desired criteria such as cell cost, inaccuracy, maximum available force and moment at the end-effector. The configuration which has the best evaluation result is considered the final optimal structure[2,3].

The structure decision method described here is based on the advantages of the CEBOT cell structure with the following simplifying assumptions:

- (1) Each cell can connect and detach automatically.
- (2) Each cell has the same geometrical size.
- (3) The cells must connect sequentially.
- (4) There are no manipulator offsets.
- (5) Rotating joints, bending joints, and sliding joints are the only available joint cells (see Fig. 17).

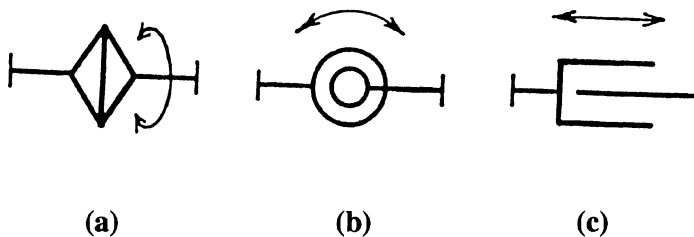


Fig. 17: Joint cells available for structure planning,
(a) rotating joint, (b) bending joint, (c) sliding joint

The task for CEBOT is divided into N work points with the following properties given for each work point:

- (1) Desired end-effector position and orientation:

$$P_n = (x_{pn}, y_{pn}, z_{pn}, \psi_{pn}, \theta_{pn}, \phi_{pn}).$$

- (2) Required force and moment at the end-effector:

$$F_n = (f_x, f_y, f_z, m_x, m_y, m_z).$$

- (3) End-effector positioning accuracy:

$$E_n = (d_x, d_y, d_z, \delta_x, \delta_y, \delta_z).$$

- (4) Manipulator base setting position and orientation:

$$Q_n = (x_{qn}, y_{qn}, z_{qn}, \psi_{qn}, \theta_{qn}, \phi_{qn}).$$

- (5) End-effector type.

- (6) Work space constraints: $S_n(x, y, z)$.

For each work point the following method leads to a table of candidates. At the base point $Q_0=Q_n$ and at the end-effector point $P_0=P_n$, two coordinate systems for each work point n with two straight lines along the z -axis of each coordinate system are defined (see Fig. 18). The line P_iQ_j is drawn with regard to the cell length l . A simple 3 link manipulator approach with the links Q_0Q_j , Q_jP_i and P_iP_0 is considered. Additional necessary degrees of freedom, joint combination, and arm link length can be found by checking the length of P_iQ_j and the angle of torsion between the three basic links. Dividing the length P_iQ_j by the cell length l , the number of cells D given by Eq.(2-2-1), which must connect between P_i and Q_j , can be calculated. For a non-integer number of cells D , an additional degree of freedom is needed (see Fig. 19). If basic links cannot satisfy the calculated angle of torsion given by equations (2-2-2,3, 4), extra degrees of freedom are needed for correction.

$$\begin{aligned}
D^2 = & \left\{ \frac{x_{p0} - x_{q0}}{l} - (n_p - \tfrac{1}{2})A_p - (n_q - \tfrac{1}{2})A_q \right\}^2 \\
& + \left\{ \frac{y_{p0} - y_{q0}}{l} - (n_p - \tfrac{1}{2})B_p - (n_q - \tfrac{1}{2})B_q \right\}^2 \\
& + \left\{ \frac{z_{p0} - z_{q0}}{l} - (n_p - \tfrac{1}{2})C_p - (n_q - \tfrac{1}{2})C_q \right\}^2,
\end{aligned}
\tag{2-2-1}$$

where

$$\begin{aligned}
A_p &= C_{\phi p} S_{\theta p} C_{\psi p} + S_{\phi p} S_{\psi p}, & A_q &= C_{\phi q} S_{\theta q} C_{\psi q} + S_{\phi q} S_{\psi q}, \\
B_p &= S_{\phi p} S_{\theta p} C_{\psi p} - C_{\phi p} S_{\psi p}, & B_q &= S_{\phi q} S_{\theta q} C_{\psi q} - C_{\phi q} S_{\psi q}, \\
C_p &= C_{\theta p} C_{\psi p}, & C_q &= C_{\theta q} C_{\psi q}, \\
S_{\theta i} &= \sin \theta_i, \quad C_{\theta i} = \cos \theta_i, \dots, & n_p, n_q &= 1, 2, 3, \dots
\end{aligned}$$

$$\mathbf{j} \cdot \overrightarrow{P_i Q_j} = 0, \tag{2-2-2}$$

$$\overrightarrow{P_0 P_i} \cdot (\overrightarrow{Q_0 Q_j} \times \overrightarrow{Q_0 P_i}) = 0, \tag{2-2-3}$$

$$\mathbf{o} \cdot \overrightarrow{P_i Q_j} = 0, \tag{2-2-4}$$

where

$$\begin{aligned}
\mathbf{j} &= (0 \ 1 \ 0), \\
\mathbf{o} &= (C_{\phi p} S_{\theta p} S_{\psi p} - S_{\phi p} C_{\psi p} \quad S_{\phi p} S_{\theta p} S_{\psi p} + C_{\phi p} C_{\psi p} \quad C_{\theta p} S_{\psi p}).
\end{aligned}$$

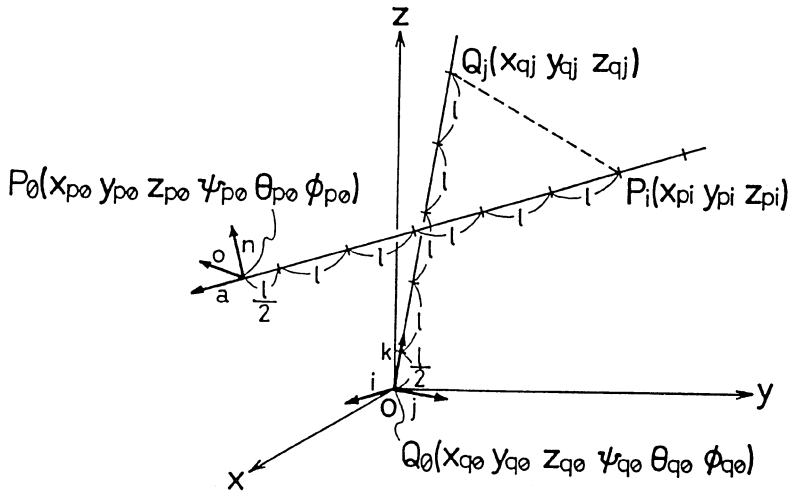


Fig. 18: Geometry for a given work point

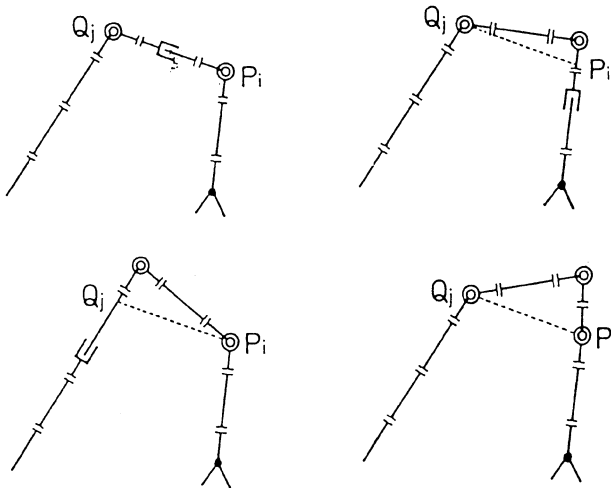


Fig. 19: Solution for a non-integer number of cells D

In this way all combinations for P_i and Q_j are checked and listed in the table of candidates. However, all candidates have to satisfy the following three geometric constraints.

Link Interference Constraint

If the links interfere with one another, the configuration must be excluded from the list of candidates. The link size is approximated by a cylinder (radius r^*) and a straight line through the link center (see Fig. 20). The distance limit between two link center lines is given as:

$$(2r^*)^2 \geq \min_{s,t} \{ (As - at + X_1 - x_1)^2 + (Bs - bt + Y_1 - y_1)^2 + (Cs - ct + Z_1 - z_1)^2 \}, \quad (2-2-5)$$

where

$$\begin{aligned} A &= X_2 - X_1, & B &= Y_2 - Y_1, & C &= Z_2 - Z_1, \\ a &= x_2 - x_1, & b &= y_2 - y_1, & c &= z_2 - z_1, \\ 0 &\leq s \leq 1, & 0 &\leq t \leq 1. \end{aligned}$$

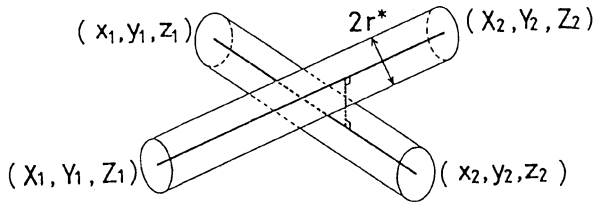


Fig. 20: Link approximation by a cylinder

Maximum Joint Movement Limit Constraint

A candidate must be excluded when exceeding the joint movement limits. Two neighbor links can be regarded as vectors, so that the angle between them is given as:

$$\theta^* = \cos^{-1} (L_1 \cdot L_2 / |L_1| |L_2|). \quad (2-2-6)$$

This angle θ^* must stay in the rotating or bending joint movement limit given as:

$$|\theta^*| \leq \theta_{\max}. \quad (2-2-7)$$

For sliding joints the limit is given as:

$$0 \leq d^* \leq d_{\max}. \quad (2-2-8)$$

Combination Infeasibility Constraint

Each basic link needs a minimal geometric space. If, for correction of a non-integer D or the angle of torsion, an additional link has to be used, there might be no space to insert additional links into the structure. Therefore a candidate must be excluded under the following conditions:

$$\begin{aligned} & (\text{length of the basic link})/l - 1 \\ & \geq (\text{number of the combining cells}). \end{aligned} \quad (2-2-9)$$

2.2.5 Structure Evaluation Function

Out of the lists of candidates for each work point, common candidates are selected and evaluated by the evaluation function given by Eq.(2-2-10).

$$S = \sum_{n=1}^N (a_1 P_n + a_2 H_n) + a_3 \sum_{n=1}^{N-1} L_n + a_4 C \quad (2-2-10)$$

- where P_n : Positioning inaccuracy for work point n ,
 H_n : Required end-effector torque for work point n ,
 L_n : Necessary joint movement to move CEBOT
 from work point n to work point $n+1$,
 C : Cell cost parameter,
 N : Total number of work points,
 a_1, a_2, a_3, a_4 : Weight parameters dependent on application.

The entire CEBOT manipulator is considered as a rigid structure and the manipulator weight, payload, and compliance of the coupling mechanism are taken into account. With differential deflection and torsion at the connecting point of cells, the positioning inaccuracy P_n is calculated. The static torque H_n required at the end-effector is calculated by considering cell weight and manipulator payload. Joint and mobile cells have to move CEBOT from work point n to work point $n+1$. The moving parameter L_n is the sum of all joint displacements and mobile cell travel distances. The cell cost C is the sum of the number of cells considering their individual complexity and cost (Nakagawa, 1987).

The structure decision method has been tested for efficiency with a task given by two work points (see Fig. 21). There were no obstacles in the work space and the mobile cell could move everywhere to dock with the resulting structure. Four common candidates, two fixed base and two mobile base configurations are shown in Fig. 22. Table 1 shows the evaluation function results for the four candidates. It can be realized that parameter values change independently, e.g. positioning inaccuracy for candidate 1 and 3 at work point 2. The inaccuracy of candidate 1 with a fixed base is caused by its long arm, i.e. it is inaccurate in the gravity direction, whereas candidate 3 with a mobile base has inaccuracy in the horizontal direction because of its moving mechanism.

Table 1: Evaluation function for 4 common candidates

	P	H	L	C
Candidate 1	13764	520	450	12
Candidate 2	13758	593	450	12
Candidate 3	13034	187	785	11
Candidate 4	18819	291	648	13

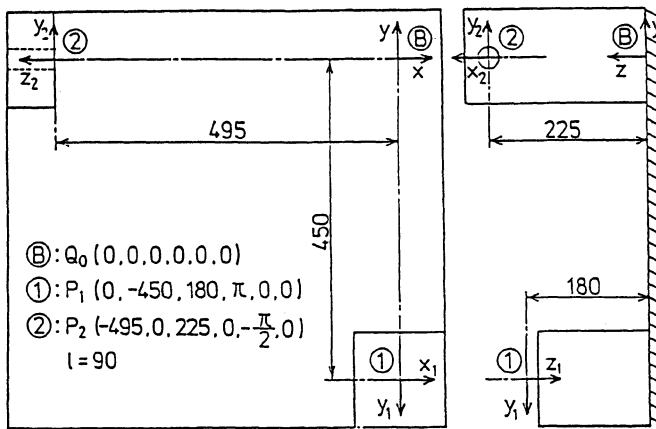
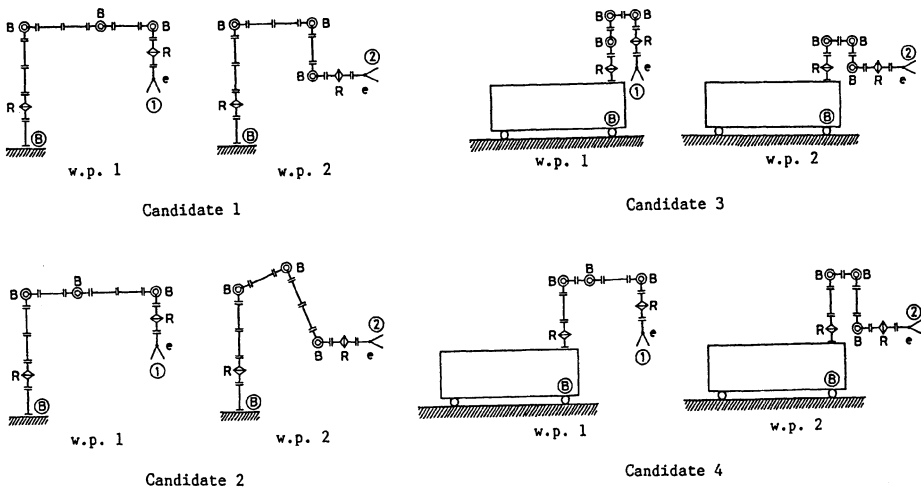


Fig. 21: Task given by two work points



w.p. : work point

Fig. 22: Result of structure decision, 4 common candidates

3. Communication System

3.1 Communication for the Undocked CEBOT

Communication in the undocked state of CEBOT is used for object cell type recognition, position, and attitude finding. A communication example is shown in Fig. 23. The moving cell distinguishes a cell with the desired function out of many object cells. Another application is the coordination of uncoupled cells or modules by communication.

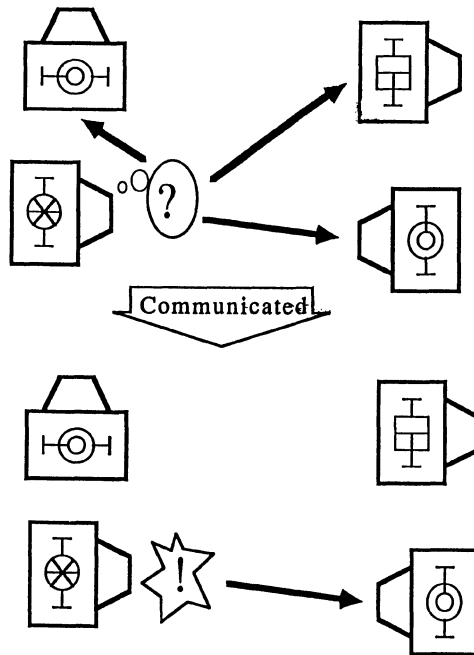


Fig. 23: Communication example

3.1.1 Communication Method

The geometric relation for cell recognition, position and attitude finding is shown in Fig. 24. Communication is achieved using an infrared LED/photodiode pair mounted in a rotating sensor module at an elevated

level (see Fig. 26). Signal transfer to the LED/photodiode is done by four brush and slip-ring sets.

The mobile cell calls for a docking partner with the desired cell function, eventually selects its out of several cells of the same type, and locates its position (see Fig. 23). This is done by a 6-step control scheme (see Table 2).

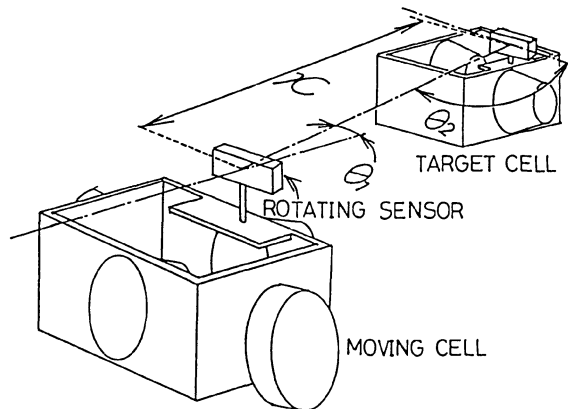


Fig. 24: Geometry of sender and receiver

Table 2: Communication sequence control scheme

Step	Digit			Description
	H	M	L	
1	1	x	F	sender x calls cell with function F object cells roughly adjust to cell x
2	2	y	x	cell y with function F answers
3	3	x	y	sender x acknowledges y
4	4	y	x	cell y sends adjustment request to x
5	5	x	y	cell x finishes adjustment and sends fine adjustment request to cell y
6	6	D1	D2	cell y finishes fine adjustment and transfers data θ_2

Digit H: Data word function **Digit M:** Transmitter-address

Digit L: Receiver-address

In step 1 of Table 2, the sender cell (mobile cell) calls an object cell with a desired function F . All object cells in the vicinity of the mobile cell receive this data, roughly adjust, and stop their sensor module to face the sender. In step 2 all cells with function F answer the call by sending their own address (y) and the sender address (x). The mobile cell (x) then takes the object cell which answered first and extracts its address (y), which is the M-digit of the answer data. To tell other cells with function F which is the communication partner, the mobile cell sends address (y) in step 3. Cells with an address unequal to (y) reset their state to attention, start rotating their sensor module again, and wait for a new request. Only cell (y) sends an adjustment request to the mobile cell (step 4). The sensor module of cell (y) is still not rotating, so the sender cell can adjust its sensor module to cell (y). When the adjustment is finished, the sender requests a second fine adjustment of cell (y) and data transfer of θ_2 (step 5). After fine adjustment, cell (y) transfers the value of θ_2 , which concludes the transfer sequence.

The distance x between the mobile and object cell is a function of light intensity, which makes x measurable by the mobile cell. Angle θ_1 is an internal sensor value of the mobile cell. Using the three parameters x , θ_1 , θ_2 , an intelligent approach, which has better performance than the previously described 5-step approach algorithm, is realizable.

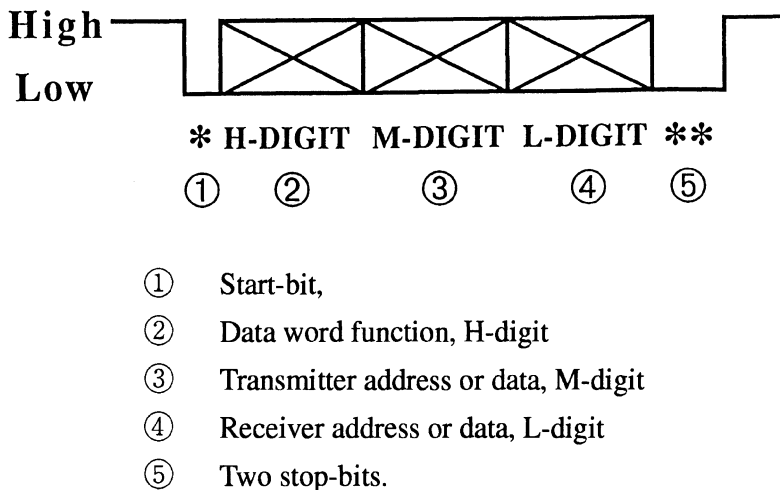


Fig. 25: Serial communication protocol using 12 bit

Communication is made using a serial 12 bit, 1 start-bit and 2 stop-bit protocol shown in Fig. 25. If the photodiode value of the receiving cell shows a negative edge after being positive a certain time, the start-bit (①) is recognized. The start-bit is followed by 4 bits (H-digit ②), which determine the function of the transferred data word. Dependent on data word function, the following two digits, M-digit (③) and L-digit (④), are either the transmitter and receiver address or the value of θ_2 (see Table 2). The two stop-bits (⑤) are followed again by a certain positive time. If the positive time at the beginning or end of the transmission misses, the whole data word becomes invalid. This error check is needed for cases like lost slip-ring contact, turn-away of the rotating sensor, etc.

3.1.2 Hardware

The rotating sensor module is used for large distances because of its high sensitivity as compared to the photodiodes PD1, PD2, PD3. However, steps 4 and 5 of the previously described approach algorithm are still used for the final fine approach.

The rotating sensor module is shown in Fig. 26 and its control system structure in Fig. 27. Motor speed and direction are controlled by a D/A-converter output of the controlling computer. The infrared LED is connected to digital input/output (DI/DO). Via a high sensitive photodiode amplifier, the photodiode is connected to an A/D-input. Position control of the rotating sensor module is done by a software control-loop using a proportional control law.



Fig.26: Photo of the rotating sensor module

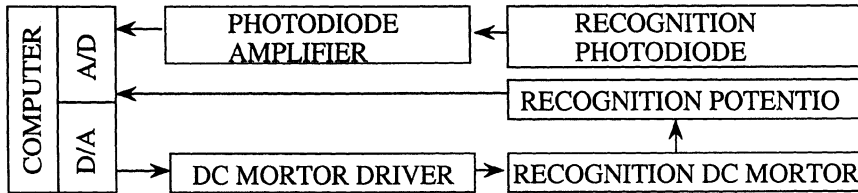


Fig.27: Control system for the rotating sensor module

3.1.3 Experimental Results

Figure 28 shows a signal timing of the rotating sensor LED and photodiode, where the LED status is inverted (low=LED on). The setting of mobile and object cells for a communication experiment is shown in Fig. 29. The mobile cell (address=0) calls for the bending joint cell (cell function F=3, address=1). The sliding joint cell (cell function F=5, address=2) does not answer the call because of its different function. The mobile cell 0 communicates with cell 1 and displays the three parameters x , θ_1 , θ_2 (see Fig. 30), which are used as input parameters for an intelligent approach algorithm including path planning.

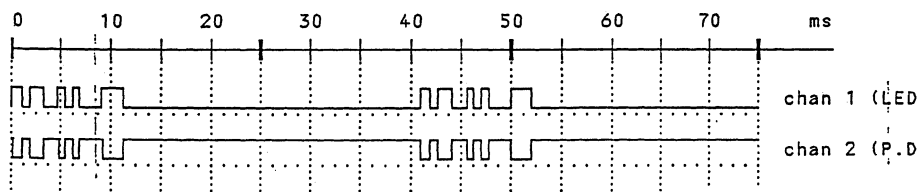


Fig. 28: Signal timing diagram

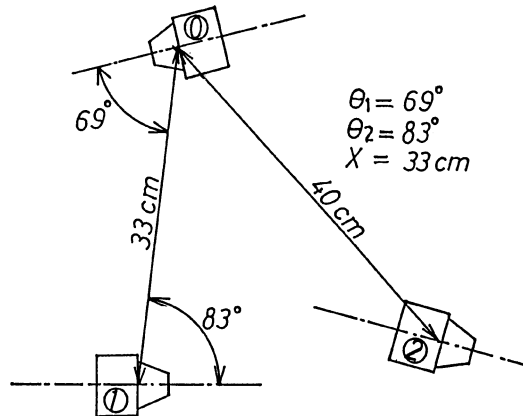


Fig. 29: Setting of mobile and object cells

```

1 | sender ?0
   | MOVING_CELL=1
   | TARGET_CELL=2
   | BENDING_JOINT_CELL=3
2 | | ROTATING_JOINT_CELL=4
   | | SLIDING_JOINT_CELL=5
   | | function ? 1
3 | |
4 | | cell #1 rough adjustment ended.
5 | | cell #2 rough adjustment ended.
   | | data error. 808
   | | data error. 800
   | | data error. 0
6 | | cell #2 has received data 101 .
   | | ***cell #0 calls cell with function 1.
   | | cell #1 has received data 101 .
7 | | ***cell #0 calls cell with function 1.
   | | cell #2 has not the desired function!
8 | | cell #0 has sent data 101
   | | cell #0 has received data 210 .
   | | ***cell #1 with function 0 answers.
   | | cell #1 has sent data 210
9 | | cell #1 has received data 301 .
   | | ***cell #0 acknowledges receiver 1.
   | | cell #0 has sent data 301
   | | cell #0 has received data 410 .
10 | | ***cell #1 sends adjustment request to cell 0.
   | |
   | | cell #0 fine adjustment ended.
   | | cell #1 has sent data 410
   | | cell #1 has received data 501 .
11 | | ***cell #0 finishes fine adjustment and
   | | | sends fine adjustment request to cell 1.
   | |
   | | cell #1 fine adjustment ended.
   | | cell #0 has sent data 501
   | | cell #0 has received data 62f .
   | | ***cell #1 finishes fine adjustment and transfers data.
   | | Transfer sequence ended successfully!
12 | | cell #0 has received theta2 = 67.7
   | | cell #0 theta 1 = 51.4
   | | distance x = 29 cm

```

Fig. 30: Communication experimental result x , θ_1 , θ_2

3.2 Communication for the Docked CEBOT

For operation of CEBOT in the docked state, the communication of various data is performed by a communications network (see Fig. 5). An example of a given CEBOT structure of 4 coupled cells consisting of mobile cell, bending joint cell, sliding joint cell, and branching cell is shown in Fig. 31. The 4 cells are considered to be autonomous with independent CPUs. If this configuration of CEBOT has to approach and dock with an additional cell on its front side, the sensor values of the front cell are transferred to the mobile cell in the rear for the approach algorithm. Other applications are that a cell reads or changes actuator positions of other cells, checks the status of other cells, scans the entire structure of CEBOT, etc. Coordination of several mobile or joint cells in the docked state is also realizable by communication.

In the following, a communications network for these purposes is described. Using this network, every cell can exchange data with any other cell and read and alter actuator positions. A very important feature of the described method is the recognition and location of failures in the network, i.e. a fault-tolerance system.

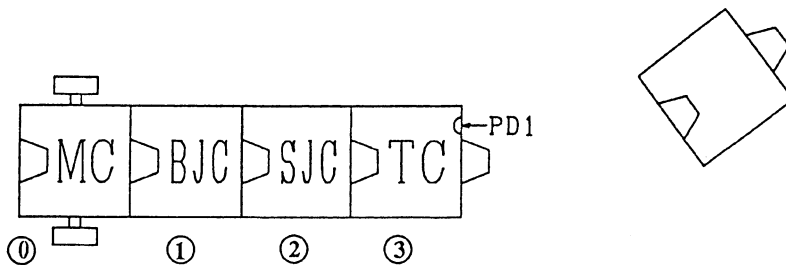


Fig. 31: Communication example for a docked CEBOT structure

3.2.1 Communication Method

All members of the communications network can control a communication sequence and can exchange data with any other cell. By

monitoring the data on the network, the cells always know always if the network is idle or busy and which cells are exchanging information.

A communication sequence has 3 steps (see Fig. 32). The master of a sequence is the cell, which starts communication and requests data exchange. In STEP 1, the master sends an attention request to the communication partner cell (slave). The slave sends back a confirmation, which signals the master that the slave cell is present and ready to communicate. If the sender does not receive this confirmation within a certain time limit, this leads to an error-check algorithm described later (see Sect. 3.2.2). STEP 2 is for data transmission, i.e. the master sends the desired data exchange sequence. The slave then sends the requested data or receives new actuator commands from the master. Finally in STEP 3, the master sends an attention terminator to the slave, which then reconfirms. This second confirmation signals the successful end of the communication sequence to the master. All other cells now know, that the communication network is idle and can they now become master and initiate a new communication sequence.

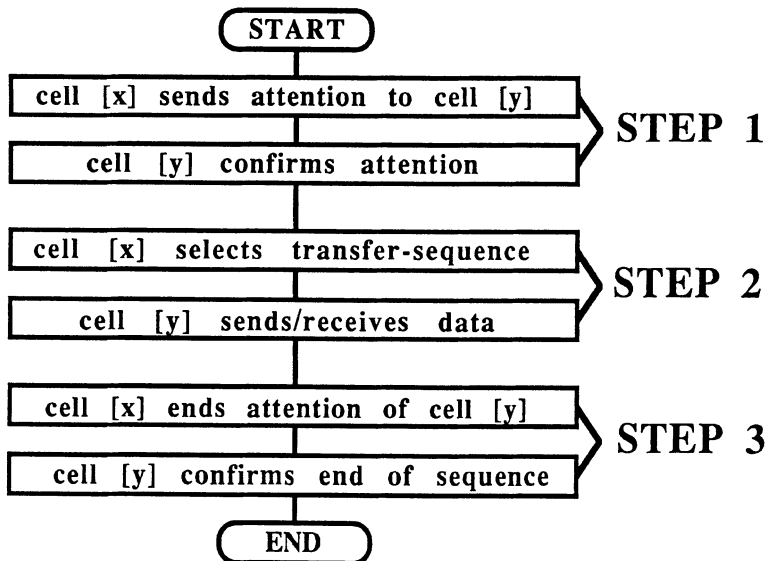


Fig. 32: 3-STEP handshake communication sequence

For data transfer on the communication network, an 8-bit organized transfer protocol is used, which is universally applicable and independent

of hardware or data-carrier. It is possible to implement this protocol on a parallel bus structure, on parallel/serial combinations, for serial transfer or with synchronous as well as asynchronous systems. The data-carrier has only to be capable of transferring 8-bit of data, where one bit is used as a valid data signal. When this bit goes negative, the whole data word is valid. In case of parallel bus implementation, this data-valid signal guarantees signal stability. For serial implementations the data valid bit is the same as an additional start-bit. In synchronous systems with a fixed data transfer rate, it signals validity of the transferred data, i.e. when positive, the transmitted data is useless.

Three bus function bits of the transferred 8-bit data word determine the word type and 4 bits are essentially transferred data. Using this convention, in one data transfer cycle, one data digit of 8 different types can be transferred. The 8 types of transferred data are shown in Table 3, i. e. control-, data-, address- higher/lower digit, and bus-reset. The type of transferred data is always identifiable by all members of the network.

Table 3: Bus function bits

Bit 2 1 0	Bus function, data word type
0 0 0	Data low digit
0 0 1	Data high digit
0 1 0	Address low digit
0 1 1	Address high digit
1 0 0	Control low digit
1 0 1	Control high digit
1 1 0	spare
1 1 1	bus reset

In the following description, words are 8-bit sized and composed of 2 digits transferred in 2 consecutive transfer cycles. The first transfer cycle is the higher, and the second cycle the lower digit.

Address words are used to address cells in the network. The cell in the rear of CEBOT is assigned the address 0, which is incremented by 1 for each neighbor-cell moving forward (see Fig.31). The front of the CEBOT is designated by the convex side of the hook-type coupling mechanism.

Data words are used to transfer sensor values, actuator positions, etc.

Control words invoke communication sequences, i.e. command another cell to attention, select the kind of data to be transferred, check the status of cells, etc. A list of control words and corresponding functions is shown in Table 4. With 256 possible control-codes any desired function, even for different robot hardware, is realizable.

Table 4: List of control words and their function

Ctrl	Description
\$0x	Confirmation from cell [x]
\$10	Attention, next word is address
\$11	End of Attention, next word address
\$12	Check for cell malfunction
\$20	Read value of left photodiode
\$21	Read value of center photodiode
\$22	Read value of right photodiode
\$23	Read value of rotating photodiode
\$24	Read obstacle sensor, left receiver
\$25	Read obstacle sensor, right receiver
\$3z	Read present actuator position
\$4z	Change actuator to new position
	z=0 -> hook coupling mechanism
	z=1 -> joint actuator
	z=2 -> rotating sensor motor
	z=3 -> obstacle sensor, sender
	z=4 -> LEDs for attitude finding

3.2.2 Error Detection

For fault-tolerance checks of the communication network, it is very important to have error recognizing capability. Error detection for the described communication method is realizable by checking transferred words for their type and checking the word order and number of cycles in a transfer sequence. The handshake based 3-step transfer sequence always insures, that the receiving cell is present and ready to receive data. Using a double confirmation signal, it is possible to recognize the presence of the receiver and the end of the transfer sequence independently.

In the actual CEBOT implementation, there is a time-out check for the confirmation handshake signal, which, if not received, leads to an error check algorithm. This error check is able to localize the failure point, i.e. connection failure, cell malfunction, etc., by performing a structure scan and cell status check.

3.2.3 CEBOT Cell Check and Structure Scan

The cell which detects an error or wants to scan the structure checks other cells by sending an attention message. If no confirmation is returned, the algorithm decides between the absence or the malfunction of the cells, which did not confirm. If absence is decided, upon it is still not known if the cell is really absent or only if the connection failed. This can be checked by separating the structure at the supposed failure point and moving the intact part of CEBOT slightly backwards. Presence or absence can then be checked by the obstacle detection sensor of the front cell, or by the rotating sensor module and communication method described in Sect. 3.1.

3.2.4 Hardware

The CEBOT hardware has a 14-pin connector included in the hook-type coupling mechanism (see Fig. 33). The communication bus (COMBUS) is realized as parallel bus architecture.

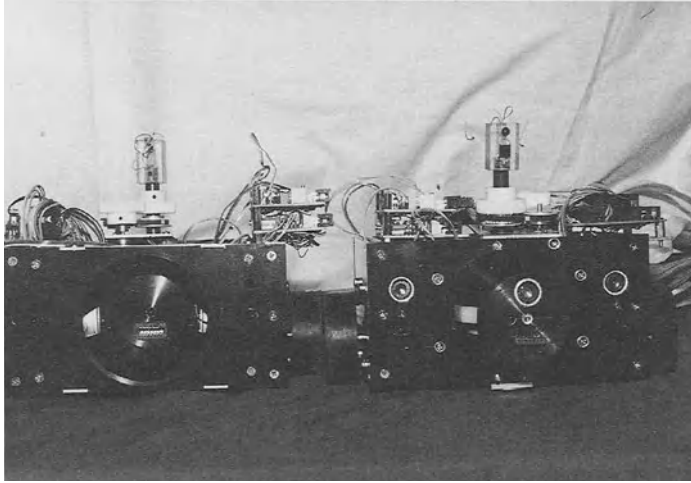


Fig. 33: Photo of the 14-pin connector for communication

3.2.5 Experimental Results

Figure 34 shows the error-free case of a COMBUS timing diagram for data transfer of photodiode PD1 value from the front cell (address=3) to the mobile cell in the back (address=0). In Fig.35 no confirmation signal was received which led to the above described cell check and structure scan.

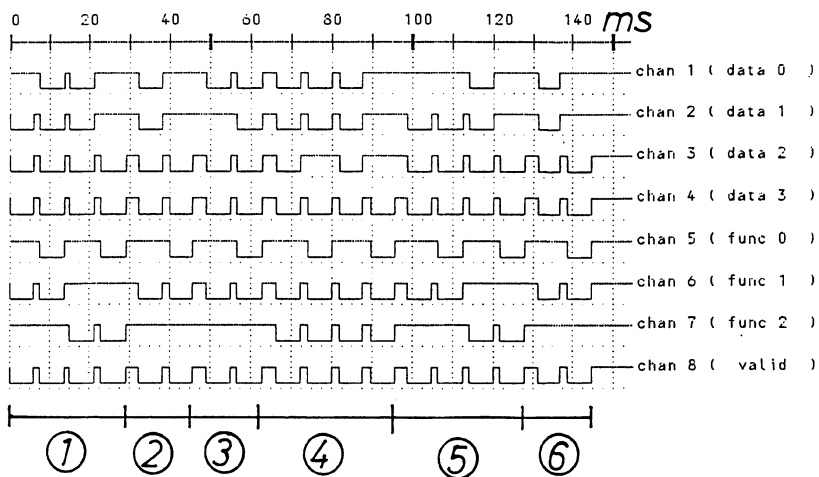


Fig. 34: COMBUS timing diagram, error-free

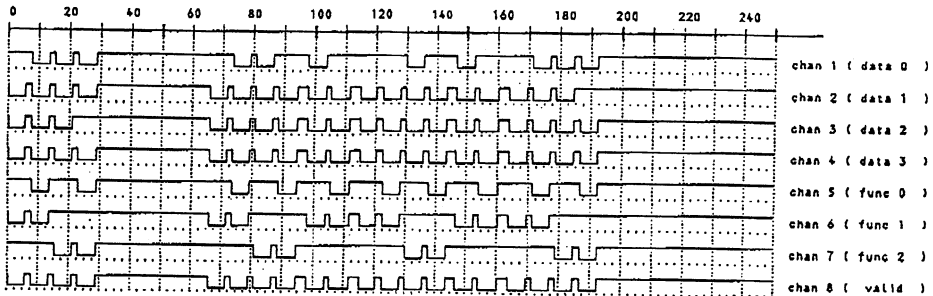


Fig. 35: COMBUS timing diagram, no confirmation error

4. Universally Applicable End-Effector Construction by CEBOT

The expert system should generate an end-effector by using the cell characteristics of the **Cellular Robotic System CEBOT**. Cells are generally much smaller than objects to be handled. The knowledge base includes object data (e.g. bolts and tools) which can be built by CEBOT (e.g. adjustable and normal spanners, screw-drivers, pliers, pincers, etc). Extraordinary techniques like hammers or other tools should also be considered for handling certain objects. By using knowledge about objects and tools, an end-effector structure can be generated by the construction rules in the rule-base.

4.1 Knowledge Representation

The representation for objects must be suitable for a range of geometric forms, sizes, weights, forces, etc. An example of a typical bolt representation is shown in the upper part of Fig. 36. A bolt has a geometry, i.e. size and shape and these parameters should be implemented

not as explicit numbers, but in descriptive terms such as "big", "small", "round", "quadratic", "ball", etc. The same holds true for the torque necessary to fasten or loosen the bolt and its holding force. Here a required moment range or magnitude order of the holding force should be used. Bolt types include nuts, screws, cotter pins, etc., which have to be turned or pushed.

For many assembly, construction or repair tasks a great variety of end-effector tools are available. These tools are usually only for one special purpose and most of them, like spanners or others, are only for fixed sizes. Heuristic knowledge of these tools has been compiled over many years and should be used in the data representation of tools and in the construction rule base. The lower part of Fig. 36 shows a knowledge representation for manufacturing tools. As with object, descriptions sizes should not be given by explicit data, but as ranges and magnitudes. Many functions are available using conventional tools like nippers, screw-drivers, etc. These tools may be used in a variety of spatial configurations with regard to the object.

The rule-base must have rules to determine an end-effector construction for the given object. Heuristic knowledge of tools has to be combined with empirical, universally applicable methods to find a solution for handling the given object. The expert system has to search in the knowledge trees for coincident data or magnitude orders of object and tools. It then decides which of the available end-effector tool types is the best for the object. In this way end-effector configurations for CEBOT are generated.

4.2 End-Effector Examples

Fig. 37 shows some examples of CEBOT end-effectors handling some objects. Any type of end-effector for handling from the side, above, etc., is realizable.

4.3 Expert System Structure

The structure of the expert system is shown in Fig. 38. The task for the expert system is given by the user. The system then analyzes and decomposes the task in the task planning level. The task execution level structures it to the cell structure of CEBOT, checks the availability of hardware, and considers geometric and other constraints. Finally, the expert system outputs control commands for the execution procedures, i.e. commands for docking, coordination, module and structure set-up, etc.

The system should also have the capacity to learn new methods, end-effector types and objects. Automatic knowledge acquisition would also be very desirable.

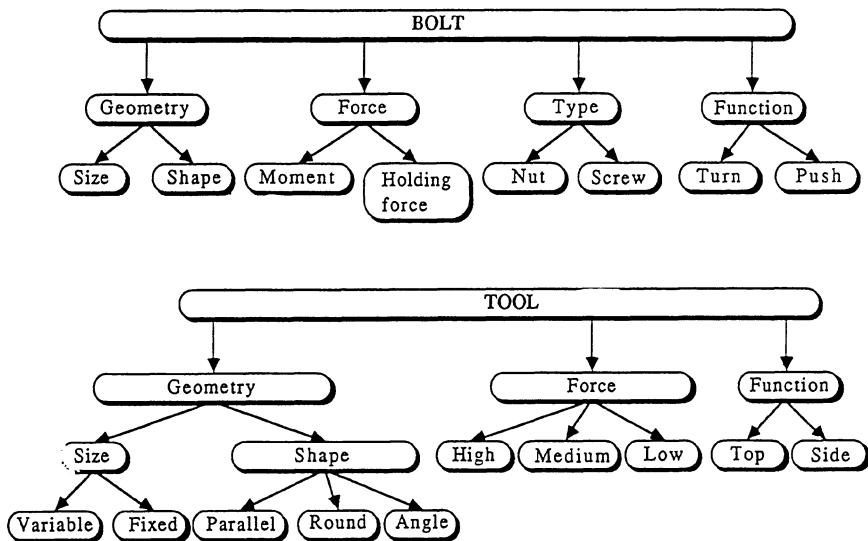


Fig. 36: Knowledge representation

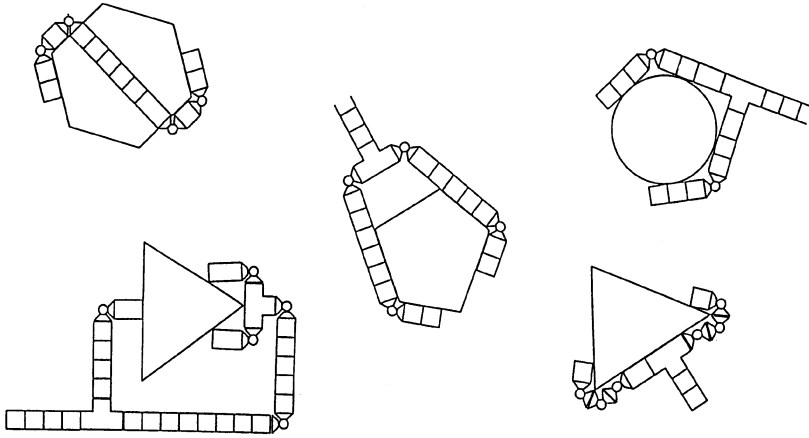


Fig. 37: End-effector examples constructed by CEBOT

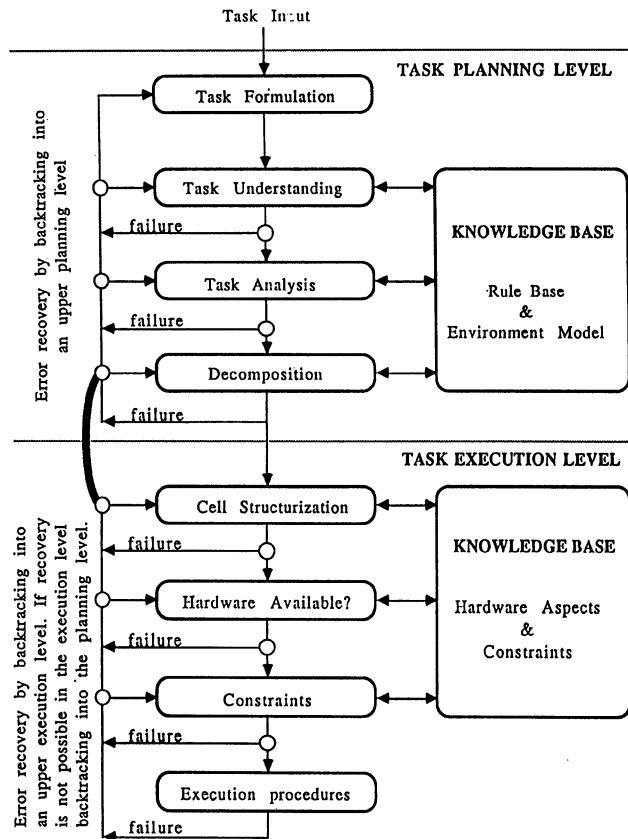


Fig. 38: Expert system structure

5. Conclusion

The idea of **Dynamically Reconfigurable Robotic Systems(DRRS)** and its realization as a **Cellular Robotic System(CEBOT)** have been described in this report. The analogy with biological systems has been outlined muscle/hardware, nerve/communication and brain/intelligence systems.

In Sect. 2 the existing CEBOT series II hardware and concepts, including automatic approach, connection, separation, structure planning, and structure evaluation have been described. Automatic approaches are conducted efficiently by the series II CEBOT. Simulations showed the effectiveness of the structure generation and evaluation methods.

Section 3 showed two communication methods for the undocked and docked state of CEBOT. Communications are used for cell type recognition and position finding before docking. Using the communication method to recognize cell distance and relative position, a more efficient approach can be realized. The method for the docked state of CEBOT described in Sect. 4 is for system operation, sensor data exchange, actuator position alteration, and malfunction detection. The communication network is based on 8-bit parallel data transmission and with 256 communication sequences available it is widely applicable. It might be applied to other robot hardware or other 8-bit data-carriers. Experimental results for the correct and error case have been shown. In case of errors during communication, the cause and point of failure can be detected by an error-check program.

The concept of an expert system for an universally applicable end-effector built by CEBOT was presented. The object and tool knowledge representation must be described by orders of magnitude and not specific numbers. Knowledge trees for objects (e.g. bolts) and manufacturing tools have been described. End-effector examples illustrate the idea of universally applicable end-effector configurations.

This cellular robotic system will be implemented under other hostile environments, such underwater and gas contained environments. This system is also applicable for bio-medical purposes based on the micro-electronic mechanical system. The small cells can be made by the IC technology, thus these cells can be sent to the biological system/

human system to inspect and make operation from the inside of these systems.

The energy resource is then one of most critical problems in the Cellular Robotic System. Energy cell has to be built to distribute powers to the whole system. The alternative will be the power supply by the non-contact manner from the outside. This will be a future problem in the Cellular Robotic System.

References

1. T. Fukuda, S. Nakagawa, "Dynamically Reconfigurable Robotic System", Proceedings of IECON'87, pp.588-595, 1987.
2. T. Fukuda, S. Nakagawa, Y. Kawauchi, M. Buss, "Self Organizing Robots Based on Cell Structures - CEBOT", Proceedings, IEEE International Workshop on Intelligent Robots and Systems (IROS'88), Science Univ. of Tokyo, pp.145-150, 1988.
3. T. Fukuda, S. Nakagawa, Y. Kawauchi, M. Buss, "A structure decision method for self organizing robots based on cell structures - CEBOT", 1989 IEEE International Conference on Robotics and Automation, to appear 1989.

List of Participants

Patrick Aebischer
Centre Hospitalier Univ. Valdois
Division Autonome de
Recherche Chirurgicale
1011 Lausanne, Switzerland

Masuo Aizawa
Tokyo Institute of Technology
Dept. of Bioengineering
4259 Nagatsuta, Midori-ku
Yokohama 227 Japan

James Albus
NIST National Institute of
Standards and Technology
Robot Systems Division
Bldg. 220, Room B124
Gaithersburg, MD 20899, U.S.A.

Ruzena Bajcsy
University of Pennsylvania
School of Eng. and Information Sc.
GRASP Lab
3401 Walnut Street, Rm 301C
Philadelphia, PA 19104-6228, U.S.A.

Dana H. Ballard
University of Rochester
Computer Science Dept.
Rochester, NY 14627, U.S.A.

Gerardo Beni
University of California
College of Engineering
200 University Office Building
Riverside, CA 92521-0425, U.S.A.

Massimo Bergamasco
Scuola Superiore S. Anna
ARTS Lab
Via Carducci, 40
56100 Pisa, Italy

Emilio Bizzi
M.I.T.
Dept. of Brain and Cognitive Science
Cambridge, MA 02139, U.S.A.

Stanley J. Bolanowski
Syracuse University
Institute for Sensory Research
Syracuse, NY 13244-5290, U.S.A.

Tom Bosser
Universität Münster
Institut für Allgemeine &
Angewandte Psychologie
D-4400 Munster, FRG

Martin Brooks
NRC - CNRC
Inst. for Information Technology
Bldg. 50
Ottawa K1A 0R6, Canada

Rodney A. Brooks
M.I.T. - AI Lab
545 Technology Square
Cambridge, MA 02139, U.S.A.

David Burr
Istituto di Neurofisiologia
del C.N.R.
Via S. Zeno, 31
56100 Pisa, Italy

Giuseppe Casolino
University of Genoa
D.I.S.T.
Via Opera Pia 11a
16145 Genoa, Italy

Alice M. Chiang
M.I.T.
Lincoln Laboratory
Lexington, MA 02173-0073, U.S.A.

Paolo Dario
Scuola Superiore S. Anna
ARTS Lab
Via Carducci, 40
56100 Pisa, Italy

Danilo De Rossi
University of Pisa
Centro E. Piaggio
Via Diotallevi, 2
56100 Pisa, Italy

Shimon Edelman
The Weizmann Inst. of Science
Dept. of Applied Mathematics
and Computer Science
Rehovot 76100, Israel

Fabrizio Ferrari
University of Genoa
D.I.S.T.
Via Opera Pia 11a
16145 Genoa, Italy

Tamar Flash
The Weizmann Institute of Science
Dept. of Applied Mathematics
and Computer Science
Rehovot 76100, Israel

Anita Flynn
M.I.T. - AI Lab
545 Technology Square
Cambridge, MA 02139, U.S.A.

Toshio Fukuda
Nagoya University
Dept. Mechano-Informatics
Furo-cho, Chikusa-ku
Nagoya 46401 Japan

Stefano Gentilucci
University of Parma
Istituto di Fisiologia Umana
Via Gramsci, 14
43100 Parma, Italy

Massimo Grattarola
University of Genoa
D.I.B.E.
Via Opera Pia 11a
16145 Genoa, Italy

Blake Hannaford
University of Washington
Dept. of Electrical Eng., FT-10
Seattle, WA 98195, U.S.A.

Vincent Hayward
McGill University
Dept. of Electrical Engineering
3480 University Street
Montreal, Quebec, H3A 2A Canada

Neville Hogan
M.I.T.
Dept. of Mechanical Engineering
77 Massachusetts Avenue, RM 3-449D
Cambridge, MA 02139, U.S.A.

Gideon F. Inbar
Technion Israel Inst. of Technology
Dept. of Electrical Engineering
Technion City
Haifa 32000, Israel

Michael I. Jordan
M.I.T.
Dept. Brain and Cognitive Sciences
E10-034D
Cambridge, MA 02139, U.S.A.

Roland G. Julian
Air Force Aerospace Medical Res. Lab.
Biodynamics and Bioengineering Div.
Wright-Patterson Air Base, OH 45433, U.S.A.

M. Okay Kaynak
Bogazici University
Electrical and Electronic
Engineering Dept.
Istanbul, Turkey

Jeff Koechling
Cornell University
Sibley School of Mechanical Eng.
Ithaca, NY 14853-7501, U.S.A.

Gregory T.A. Kovacs
Stanford University
Center for Integrated Systems
Room AEL 132
Stanford, CA 94305-4070, U.S.A.

Lina Massone
Northwestern University
Dept. of Physiology, Word 5-198
303 E. Chicago Avenue
Chicago, IL 60611, U.S.A.

Claudio Melchiorri

University of Bologna
D.E.I.S.
Viale Risorgimento, 2
40136 Bologna, Italy

B. G. Mertzios

Democritus University of Thrace
Automatic Control Systems Lab.
Dept. of Electrical Engineering
67100 Xanthi, Greece

Alex Meystel

Drexel University
College of Engineering
Dept. of Electrical and Computer Eng.
Philadelphia, PA 19104, U.S.A.

Jonathan Monsarrat

3 Ames Street
Cambridge, MA 02139, U.S.A.

Adam Morecki

Inst. Aircraft Eng. & Appl. Mechanics
Warsaw University of Technology
Al. Niepodleglosci 222
00-665 Warszawa, Poland

Maria Concetta Morrone

Istituto di Neurofisiologia
del C.N.R.
Via S. Zeno, 31
56100 Pisa, Italy

Ferdinando Mussa Ivaldi

M.I.T.
Dept. of Brain and Cognitive Sc.
Cambridge, MA 02139, U.S.A.

Krishna C. Persaud

UMIST/DIAS
P.O. Box 88
Seakville Street
Manchester M60 1QD, U.K.

Giacomo Rizzolatti

University of Parma
Istituto di Fisiologia Umana
Via Gramsci, 14
43100 Parma, Italy

Joseph M. Rosen

Stanford University
Department of Surgery
Stanford, CA 94305, U.S.A.

Angelo Sabatini

Scuola Superiore S. Anna
ARTS Lab
Via Carducci 40
56100 Pisa, Italy

Giulio Sandini

University of Genoa
D.I.S.T.
Via Opera Pia 11a
16145 Genoa, Italy

Eric Schwartz

New York University Medical Center
Computational Neuroscience Labs
550 First Avenue
New York, NY 10012, U.S.A.

Atsuo Takanishi

Waseda University
Dept. of Mechanical Engineering
Ookubo 3-4-1, Shinjuku-ku
Tokyo 160 Japan

Kazuo Tanie

AIST-MITI
Mechanical Eng. Laboratory
Bio-Robotics Div. - Robotics Dept.
Namiki 1-2, Tsukuba-shi
Ibaraki 305 Japan

Jack Todd

Dept. of Mechanical Engineering
University of Edinburgh
King's Buildings, Mayfield Road
Edinburgh, EH9 3JL, U.K.

William T. Townsend

Barrett Technology, Inc.
545 Concord Avenue
Cambridge, MA 02138, U.S.A.

Robert F. Valentini

Brown University
Artificial Organ Laboratory
Providence, RI 02912, U.S.A.

Jan A. Van Alsté

University of Twente
Biomedical Engineering Division
P.O. Box 217
7500 AE Enschede, The Netherlands

William S. Vaughan
Office of Naval Research
Cognitive & Neural Sciences Div.
800 N. Quincy Street
Arlington, VA 22217, U.S.A.

Gabriele Vassura
University of Bologna
D.I.E.M.
Viale Risorgimento, 2
40136 Bologna, Italy

Ronald T. Verrillo
Syracuse University
Institute for Sensory Research
Merrill Lane
Syracuse, NY 13244-5290, U.S.A.

E. Von Goldammer
Institut für Biomedizintechnik
Biophysik und Biokybernetik
Nordische Universität, FRG

Harry Wechsler
George Mason University
Dept. of Computer Science
4400 University Drive
Fairfax, VA 22030, U.S.A.

James F. Wilson
Duke University
School of Engineering
Dept. of Civil and Environmental Eng.
Durham, NC 27706, U.S.A.

Uwe Windhorst
The University of Calgary
Health Sciences Centre
3330 Hospital Drive N.W.
Calgary, Alberta, Canada T2N 4N1

Renato Zaccaria
University of Genoa
D.I.S.T.
Via Opera Pia 11a
16145 Genoa, Italy

NATO ASI Series F

Including Special Programmes on Sensory Systems for Robotic Control (ROB) and on Advanced Educational Technology (AET)

Vol. 1: Issues in Acoustic Signal - Image Processing and Recognition. Edited by C. H. Chen. VIII, 333 pages. 1983.

Vol. 2: Image Sequence Processing and Dynamic Scene Analysis. Edited by T. S. Huang. IX, 749 pages. 1983.

Vol. 3: Electronic Systems Effectiveness and Life Cycle Costing. Edited by J. K. Skwirzynski. XVII, 732 pages. 1983.

Vol. 4: Pictorial Data Analysis. Edited by R. M. Haralick. VIII, 468 pages. 1983.

Vol. 5: International Calibration Study of Traffic Conflict Techniques. Edited by E. Asmussen. VII, 229 pages. 1984.

Vol. 6: Information Technology and the Computer Network. Edited by K. G. Beauchamp. VIII, 271 pages. 1984.

Vol. 7: High-Speed Computation. Edited by J. S. Kowalik. IX, 441 pages. 1984.

Vol. 8: Program Transformation and Programming Environments. Report on a Workshop directed by F. L. Bauer and H. Remus. Edited by P. Pepper. XIV, 378 pages. 1984.

Vol. 9: Computer Aided Analysis and Optimization of Mechanical System Dynamics. Edited by E. J. Haug. XXII, 700 pages. 1984.

Vol. 10: Simulation and Model-Based Methodologies: An Integrative View. Edited by T. I. Ören. B. P. Zeigler, M. S. Elzas. XIII, 651 pages. 1984.

Vol. 11: Robotics and Artificial Intelligence. Edited by M. Brady, L. A. Gerhardt, H. F. Davidson. XVII, 693 pages. 1984.

Vol. 12: Combinatorial Algorithms on Words. Edited by A. Apostolico, Z. Galil. VIII, 361 pages. 1985.

Vol. 13: Logics and Models of Concurrent Systems. Edited by K. R. Apt. VIII, 498 pages. 1985.

Vol. 14: Control Flow and Data Flow: Concepts of Distributed Programming. Edited by M. Broy. VIII, 525 pages. 1985.

Vol. 15: Computational Mathematical Programming. Edited by K. Schittkowski. VIII, 451 pages. 1985.

Vol. 16: New Systems and Architectures for Automatic Speech Recognition and Synthesis. Edited by R. De Mori, C.Y. Suen. XIII, 630 pages. 1985.

Vol. 17: Fundamental Algorithms for Computer Graphics. Edited by R. A. Earnshaw. XVI, 1042 pages. 1985.

Vol. 18: Computer Architectures for Spatially Distributed Data. Edited by H. Freeman and G. G. Pieroni. VIII, 391 pages. 1985.

Vol. 19: Pictorial Information Systems in Medicine. Edited by K. H. Höhne. XII, 525 pages. 1986.

Vol. 20: Disordered Systems and Biological Organization. Edited by E. Bienenstock, F. Fogelman Soulié, G. Weisbuch. XXI, 405 pages. 1986.

Vol. 21: Intelligent Decision Support in Process Environments. Edited by E. Hollnagel, G. Mancini, D. D. Woods. XV, 524 pages. 1986.

NATO ASI Series F

Including Special Programmes on Sensory Systems for Robotic Control (ROB) and on Advanced Educational Technology (AET)

Vol. 22: Software System Design Methods. The Challenge of Advanced Computing Technology. Edited by J. K. Skwirzynski. XIII, 747 pages. 1986.

Vol. 23: Designing Computer-Based Learning Materials. Edited by H. Weinstock and A. Bork. IX, 285 pages. 1986.

Vol. 24: Database Machines. Modern Trends and Applications. Edited by A. K. Sood and A. H. Qureshi. VIII, 570 pages. 1986.

Vol. 25: Pyramidal Systems for Computer Vision. Edited by V. Cantoni and S. Levialdi. VIII, 392 pages. 1986. (ROB)

Vol. 26: Modelling and Analysis in Arms Control. Edited by R. Avenhaus, R. K. Huber and J. D. Kettelle. VIII, 488 pages. 1986.

Vol. 27: Computer Aided Optimal Design: Structural and Mechanical Systems. Edited by C. A. Mota Soares. XIII, 1029 pages. 1987.

Vol. 28: Distributed Operating Systems. Theory und Practice. Edited by Y. Paker, J.-P. Banatre and M. Bozyigit. X, 379 pages. 1987.

Vol. 29: Languages for Sensor-Based Control in Robotics. Edited by U. Rembold and K. Hörmann. IX, 625 pages. 1987. (ROB)

Vol. 30: Pattern Recognition Theory and Applications. Edited by P. A. Devijver and J. Kittler. XI, 543 pages. 1987.

Vol. 31: Decision Support Systems: Theory and Application. Edited by C. W. Holsapple and A. B. Winston. X, 500 pages. 1987.

Vol. 32: Information Systems: Failure Analysis. Edited by J. A. Wise and A. Debons. XV, 338 pages. 1987.

Vol. 33: Machine Intelligence and Knowledge Engineering for Robotic Applications. Edited by A. K. C. Wong and A. Pugh. XIV, 486 pages. 1987. (ROB)

Vol. 34: Modelling, Robustness and Sensitivity Reduction in Control Systems. Edited by R.F. Curtain. IX, 492 pages. 1987.

Vol. 35: Expert Judgment and Expert Systems. Edited by J. L. Mumpower, L. D. Phillips, O. Renn and V. R. R. Uppuluri. VIII, 361 pages. 1987.

Vol. 36: Logic of Programming and Calculi of Discrete Design. Edited by M. Broy. VII, 415 pages. 1987.

Vol. 37: Dynamics of Infinite Dimensional Systems. Edited by S.-N. Chow and J. K. Hale. IX, 514 pages. 1987.

Vol. 38: Flow Control of Congested Networks. Edited by A. R. Odoni, L. Bianco and G. Szegö. XII, 355 pages. 1987.

Vol. 39: Mathematics and Computer Science in Medical Imaging. Edited by M. A. Viergever and A. Todd-Pokropek. VIII, 546 pages. 1988.

Vol. 40: Theoretical Foundations of Computer Graphics and CAD. Edited by R. A. Earnshaw. XX, 1246 pages. 1988.

Vol. 41: Neural Computers. Edited by R. Eckmiller and Ch. v. d. Malsburg. XIII, 566 pages. 1988.

NATO ASI Series F

Including Special Programmes on Sensory Systems for Robotic Control (ROB) and on Advanced Educational Technology (AET)

Vol. 42: Real-Time Object Measurement and Classification. Edited by A. K. Jain. VIII, 407 pages. 1988. (ROB)

Vol. 43: Sensors and Sensory Systems for Advanced Robots. Edited by P. Dario. XI, 597 pages. 1988. (ROB)

Vol. 44: Signal Processing and Pattern Recognition in Nondestructive Evaluation of Materials. Edited by C. H. Chen. VIII, 344 pages. 1988. (ROB)

Vol. 45: Syntactic and Structural Pattern Recognition. Edited by G. Ferraté, T. Pavlidis, A. Sanfeliu and H. Bunke. XVI, 467 pages. 1988. (ROB)

Vol. 46: Recent Advances in Speech Understanding and Dialog Systems. Edited by H. Niemann, M. Lang and G. Sagerer. X, 521 pages. 1988.

Vol. 47: Advanced Computing Concepts and Techniques in Control Engineering. Edited by M. J. Denham and A. J. Laub. XI, 518 pages. 1988.

Vol. 48: Mathematical Models for Decision Support. Edited by G. Mitra. IX, 762 pages. 1988.

Vol. 49: Computer Integrated Manufacturing. Edited by I. B. Turksen. VIII, 568 pages. 1988.

Vol. 50: CAD Based Programming for Sensory Robots. Edited by B. Ravani. IX, 565 pages. 1988. (ROB)

Vol. 51: Algorithms and Model Formulations in Mathematical Programming. Edited by S. W. Wallace. IX, 190 pages. 1989.

Vol. 52: Sensor Devices and Systems for Robotics. Edited by A. Casals. IX, 362 pages. 1989. (ROB)

Vol. 53: Advanced Information Technologies for Industrial Material Flow Systems. Edited by S. Y. Nof and C. L. Moodie. IX, 710 pages. 1989.

Vol. 54: A Reappraisal of the Efficiency of Financial Markets. Edited by R. M. C. Guimarães, B. G. Kingsman and S. J. Taylor. X, 804 pages. 1989.

Vol. 55: Constructive Methods in Computing Science. Edited by M. Broy. VII, 478 pages. 1989.

Vol. 56: Multiple Criteria Decision Making and Risk Analysis Using Microcomputers. Edited by B. Karpak and S. Zionts. VII, 399 pages. 1989.

Vol. 57: Kinematics and Dynamic Issues in Sensor Based Control. Edited by G. E. Taylor. XI, 456 pages. 1990. (ROB)

Vol. 58: Highly Redundant Sensing in Robotic Systems. Edited by J. T. Tou and J. G. Balchen. X, 322 pages. 1990. (ROB)

Vol. 59: Superconducting Electronics. Edited by H. Weinstock and M. Nisenoff. X, 441 pages. 1989.

Vol. 60: 3D Imaging in Medicine. Algorithms, Systems, Applications. Edited by K. H. Höhne, H. Fuchs and S. M. Pizer. IX, 460 pages. 1990.

Vol. 61: Knowledge, Data and Computer-Assisted Decisions. Edited by M. Schader and W. Gaul. VIII, 421 pages. 1990.

Vol. 62: Supercomputing. Edited by J. S. Kowalik. X, 425 pages. 1990.

Vol. 63: Traditional and Non-Traditional Robotic Sensors. Edited by T. C. Henderson. VIII, 468 pages. 1990. (ROB)

NATO ASI Series F

Including Special Programmes on Sensory Systems for Robotic Control (ROB) and on Advanced Educational Technology (AET)

Vol. 64: Sensory Robotics for the Handling of Limp Materials. Edited by P. M. Taylor. IX, 343 pages. 1990. (ROB)

Vol. 65: Mapping and Spatial Modelling for Navigation. Edited by L. F. Pau. VIII, 357 pages. 1990. (ROB)

Vol. 66: Sensor-Based Robots: Algorithms and Architectures. Edited by C. S. G. Lee. X, 285 pages. 1991. (ROB)

Vol. 67: Designing Hypermedia for Learning. Edited by D. H. Jonassen and H. Mandl. XXV, 457 pages. 1990. (AET)

Vol. 68: Neurocomputing. Algorithms, Architectures and Applications. Edited by F. Fogelman Soulié and J. Hérault. XI, 455 pages. 1990.

Vol. 69: Real-Time Integration Methods for Mechanical System Simulation. Edited by E. J. Haug and R. C. Deyo. VIII, 352 pages. 1991.

Vol. 70: Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms. Edited by G. H. Golub and P. Van Dooren. XIII, 729 pages. 1991.

Vol. 71: Expert Systems and Robotics. Edited by T. Jordanides and B. Torby. XII, 744 pages. 1991.

Vol. 72: High-Capacity Local and Metropolitan Area Networks. Architecture and Performance Issues. Edited by G. Pujolle. X, 536 pages. 1991.

Vol. 73: Automation and Systems Issues in Air Traffic Control. Edited by J. A. Wise, V. D. Hopkin and M. L. Smith. XIX, 594 pages. 1991.

Vol. 74: Picture Archiving and Communication Systems (PACS) in Medicine. Edited by H. K. Huang, O. Ratib, A. R. Bakker and G. Witte. XI, 438 pages. 1991.

Vol. 75: Speech Recognition and Understanding. Recent Advances, Trends and Applications. Edited by P. Laface and Renato De Mori. XI, 559 pages. 1991.

Vol. 76: Multimedia Interface Design in Education. Edited by A. D. N. Edwards and S. Holland. XIV, 216 pages. 1992. (AET)

Vol. 77: Computer Algorithms for Solving Linear Algebraic Equations. The State of the Art. Edited by E. Spedicato. VIII, 352 pages. 1991.

Vol. 78: Integrating Advanced Technology into Technology Education. Edited by M. Hacker, A. Gordon and M. de Vries. VIII, 185 pages. 1991. (AET)

Vol. 79: Logic, Algebra, and Computation. Edited by F. L. Bauer. VII, 485 pages. 1991.

Vol. 80: Intelligent Tutoring Systems for Foreign Language Learning. Edited by M. L. Swartz and M. Yazdani. IX, 347 pages. 1992. (AET)

Vol. 81: Cognitive Tools for Learning. Edited by P. A. M. Kommers, D. H. Jonassen and J. T. Mayes. X, 278 pages. 1992. (AET)

Vol. 82: Combinatorial Optimization. New Frontiers in Theory and Practice. Edited by M. Akgül, H. W. Hamacher and S. Tüfekçi. XI, 334 pages. 1992.

Vol. 83: Active Perception and Robot Vision. Edited by A. K. Sood and H. Wechsler. IX, 756 pages. 1992.

NATO ASI Series F

Including Special Programmes on Sensory Systems for Robotic Control (ROB) and on Advanced Educational Technology (AET)

Vol. 84: Computer-Based Learning Environments and Problem Solving. Edited by E. De Corte, M. C. Linn, H. Mandl and L. Verschaffel. XVI, 488 pages. 1992. (AET)

Vol. 85: Adaptive Learning Environments. Foundations and Frontiers. Edited by M. Jones and P. H. Winne. VIII, 408 pages. 1992. (AET)

Vol. 86: Intelligent Learning Environments and Knowledge Acquisition in Physics. Edited by A. Tiberghien and H. Mandl. VIII, 285 pages. 1992. (AET)

Vol. 87: Cognitive Modelling and Interactive Environments. With demo diskettes (Apple and IBM compatible). Edited by F. L. Engel, D. G. Bouwhuis, T. Bösser and G. d'Ydewalle. IX, 311 pages. 1992. (AET)

Vol. 88: Programming and Mathematical Method. Edited by M. Broy. VIII, 428 pages. 1992.

Vol. 89: Mathematical Problem Solving and New Information Technologies. Edited by J. P. Ponte, J. F. Matos, J. M. Matos and D. Fernandes. XV, 346 pages. 1992. (AET)

Vol. 90: Collaborative Learning Through Computer Conferencing. Edited by A. R. Kaye. X, 260 pages. 1992. (AET)

Vol. 91: New Directions for Intelligent Tutoring Systems. Edited by E. Costa. X, 296 pages. 1992. (AET)

Vol. 92: Hypermedia Courseware: Structures of Communication and Intelligent Help. Edited by A. Oliveira. X, 241 pages. 1992. (AET)

Vol. 93: Interactive Multimedia Learning Environments. Human Factors and Technical Considerations on Design Issues. Edited by M. Giardina. VIII, 254 pages. 1992. (AET)

Vol. 94: Logic and Algebra of Specification. Edited by F. L. Bauer, W. Brauer and H. Schwichtenberg. 1993.

Vol. 95: Comprehensive Systems Design: A New Educational Technology. Edited by C. M. Reigeluth, B. H. Banathy and J. R. Olson. IX, 437 pages. 1993. (AET)

Vol. 96: New Directions in Educational Technology. Edited by E. Scanlon and T. O'Shea. VIII, 251 pages. 1992. (AET)

Vol. 97: Advanced Models of Cognition for Medical Training and Practice. Edited by D. A. Evans and V. L. Patel. XI, 372 pages. 1992. (AET)

Vol. 98: Medical Images: Formation, Handling and Evaluation. Edited by A. E. Todd-Pokropek and M. A. Viergever. IX, 700 pages. 1992.

Vol. 99: Multisensor Fusion for Computer Vision. Edited by J. K. Aggarwal. XI, 456 pages. 1993. (ROB)

Vol. 100: Communication from an Artificial Intelligence Perspective. Theoretical and Applied Issues. Edited by A. Ortony, J. Slack and O. Stock. XII, 260 pages. 1992.

Vol. 101: Recent Developments in Decision Support Systems. Edited by C. W. Holsapple and A. B. Whinston. XI, 618 pages. 1993.

Vol. 102: Robots and Biological Systems: Towards a New Bionics? Edited by P. Dario, G. Sandini and P. Aebischer. XII, 786 pages. 1993.

NATO ASI Series F

Including Special Programmes on Sensory Systems for Robotic Control (ROB) and on Advanced Educational Technology (AET)

Vol. 103: Parallel Computing on Distributed Memory Multiprocessors. Edited by F. Özgüner, F. Erçal. VIII, 332 pages. 1993.

Vol. 104: Instructional Models in Computer-Based Learning Environments. Edited by S. Dijkstra, H. P. M. Krammer, and J. J. G. van Merriënboer. X, 510 pages. 1993. (AET)

Vol. 105: Designing Environments for Constructive Learning. Edited by T. M. Duffy, J. Lowyck, and D. H. Jonassen. VIII, 374 pages. 1993. (AET)

Vol. 106: Software for Parallel Computation. Edited by J. S. Kowalik, L. Grandinetti. IX, 363 pages. 1993.

Vol. 107: Advanced Educational Technologies for Mathematics and Science. Edited by D. L. Ferguson. XII, 749 pages. 1993. (AET)

Vol. 108: Concurrent Engineering: Tools and Technologies for Mechanical System Design. Edited by E. J. Haug. XIII, 998 pages. 1993.

Vol. 109: Advanced Educational Technology in Technology Education. Edited by A. Gordon, M. Hacker, and M. de Vries. VIII, 270 pages. 1993. (AET)

Vol. 110: Verification and Validation of Complex Systems: Human Factors Issues. Edited by J. A. Wise, V. D. Hopkin, and P. Stager. XV, 711 pages. 1993.

Vol. 111: Cognitive Models and Intelligent Environments for Learning Programming. Edited by E. Lemut, B. du Boulay, and G. Dettori. 1993. (AET)
